

0CHAIN

0Chain: A Fast, Secure, Scalable, & Free Blockchain

Tailored For High-Performance, Zero-Trust, Decentralized Storage

Saswata Basu, Thomas Austin, Siva Dirisala, & 0Chain Team

Table Of Contents

MOTIVATION	3
EXECUTIVE SUMMARY	4
1. AN INTRODUCTION TO ØCHAIN	9
2. PRODUCTS	13
2.1 ØCHAINNET	14
2.2 ØBOX	18
2.3 ØWALLET	22
3. ØCHAIN ARCHITECTURE	23
3.1 CONSENSUS PROTOCOL	24
3.2 STORAGE PROTOCOL	25
3.3 SPLIT KEY PROTOCOL	26
3.4 TOKEN REWARD PROTOCOL	26
3.5 GOVERNANCE PROTOCOL	27
3.6 OTHER PROTOCOLS	27
4. APPENDICES	
APPENDIX 1: TEAM	29
APPENDIX 2: UNDERSTANDING ØCHAIN FINALITY	30
APPENDIX 3: CONSENSUS PROTOCOL	36
APPENDIX 4: STORAGE & TOKEN REWARD PROTOCOL	38
APPENDIX 5: SPLIT KEY PROTOCOL	40

Motivation

When we started developing 0Chain in July 2017, we were driven by the idea of redefining the cloud in the context of privacy, transparency, and user control of data. In 2018, as we dove deeper into our development process, we identified a number of critical unresolved issues in the cryptocurrency space, such as: token security, token valuation metrics, governance, and token inflation. To address the aforementioned issues, we chose to conduct additional research and development. Today, we are not just redefining the cloud, but the blockchain and cryptocurrency landscape through our novel protocols.

Executive Summary

Several roadblocks exist in today's crypto market, as well as the underlying Distributed Ledger Technologies (DLT) that power them — hindering the maturation process of the blockchain industry. Some are regulatory and security concerns, some involve technical performance, while other issues are related to token economics, valuation metrics, and governance.

1. SECURITY

Regulators identify a few key shortcomings with regards to the security of crypto assets: poor wallet infrastructure, exchange transparency, and unreliable custody of assets. 0Chain is addressing all of these issues.

Today, hardware wallets are clunky and extremely hard to use for an average person, especially for daily utility. Software wallets are notoriously prone to hacking, unless you can memorize your private key. Our software-based secure wallet is the world's first to enable a 2-device authentication; it makes transactions simple and yet highly secure with just a mobile device and a laptop.

2. SPEED AND SCALABILITY

The finality for Bitcoin is about 1 hour; Ethereum is about 3-10 minutes — something that is still too slow for a micropayment transaction and verification. We aim to address this problem. 0ChainNet is a high speed blockchain with block finality achieved within .5 to 1.5 seconds (depending on network latency), highly scalable with throughput rates of over 1,000 transactions per second, and energy efficient by way of our unique Proof of Stake protocol. We have proven these results on a public test network of 10 worldwide data centers and have completed over 6 billion transactions to verify reliability.

3. ARCHITECTURE

An Ethereum node mines a block, stores the block, and stores associated unstructured data, all on the same node. This same node handles transactions and queries from the same client. This architecture makes the node very expensive, slow and unscalable. 0Chain architecture allows for separation of duties to specific designated nodes, called miners, sharders, and blobbers. Miners receive transactions from users and they generate blocks via the consensus protocol, Sharders store these blocks and respond to queries on transactions and blocks, and Blobbers store unstructured data. This architecture allows for off-the-shelf cheap hardware (making it inexpensive to support the network as a node), faster response rates, and better scalability.

ØCHAIN

4. ZCN TOKEN

The value of ZCN is mathematically related to the data stored and other services on the network, unlike other cryptocurrency.

4.1 Addressing Cost, Value, And Volatility

Expensive transaction fees and the volatility of native blockchain cryptocurrencies is an unresolved problem for today's public blockchain networks. To address this pain point, ØChain's native cryptocurrency (ZCN) is uniquely programmed as an asset-backed token.

When a user locks their ZCN, these tokens collect an "*interest*" which can be used toward payment of transactions or data services. Through this interest-bearing feature, the transaction fee is absorbed by the interest generated on the locked tokens. Additionally, ØChainNet's core service — data storage — is also enabled by locked tokens. Upon locking, a storage service can be activated and the extent of service is determined by the number of tokens locked by the user (for more details refer to section 3.2). In both of these services, the initial locked tokens are fully redeemable upon unlocking, thus facilitating free services on the network. It's free because the network mints these tokens and is part of the inflation and underlying token economics.

Storage and transaction services have a quantifiable, real world market value. Conversely, it's frequently argued that other popular native blockchain cryptocurrencies (such as BTC, ETH, etc.) in their current state lack such an economic value beyond raw speculation. This creates a challenging process for accurate price discovery of cryptocurrencies, resulting in price volatility.

ØChain has programmed value economics into the ZCN token to curb price volatility. The lower bound value of the ZCN token can be mathematically estimated based on the number of tokens locked relative to the demand or usage of data storage and transaction services (for more details refer to section 3.4).

In other words, ZCN is backed by an allocation of transaction and data storage services, thus injecting a non-speculative, integral value into the ZCN token. The asset-backed nature of ZCN can buttress its market value and reduce its price volatility unlike other popular native cryptocurrencies.

ØCHAIN

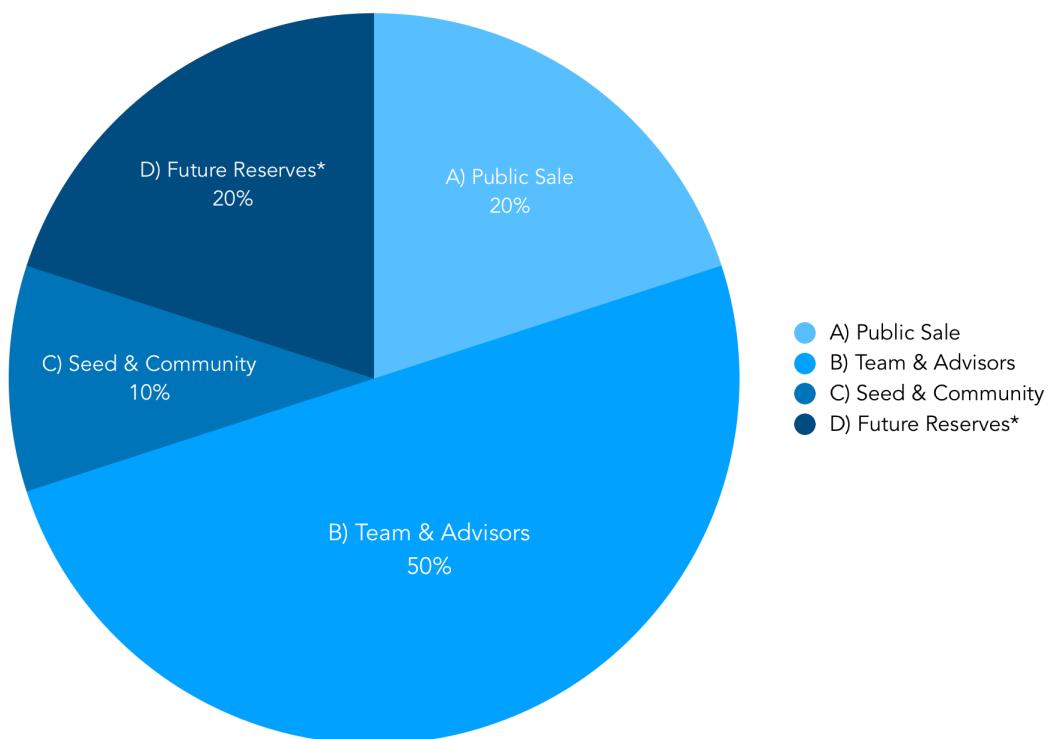
4.2 Token Supply

The ZCN token supply (at genesis block) can be broken down as follows:

- A) **Public Sale:** 40m sold to the public in the private and pre-sale rounds.
- B) **Team & Advisors:** 100m allocated to team & advisors.
- C) **Seed & Community:** 20m allocated to seed & community (eg bounties)
- D) **Future Reserves:** 40m reserved for future use if the per unit token price of ZCN exceeds \$10. This reserve unlocks in two 20m tranches in Jan 2020 and Jan 2022 if the \$10/ZCN threshold is met.

ZCN Supply Breakdown

Category	ZCN	% of Supply
A) Public Sale	40,000,000	20%
B) Team & Advisors	100,000,000	50%
C) Seed & Community	20,000,000	10%
D) Future Reserves (\$10/ZCN vesting provision)	40,000,000	20%
Total	200,000,000	100%

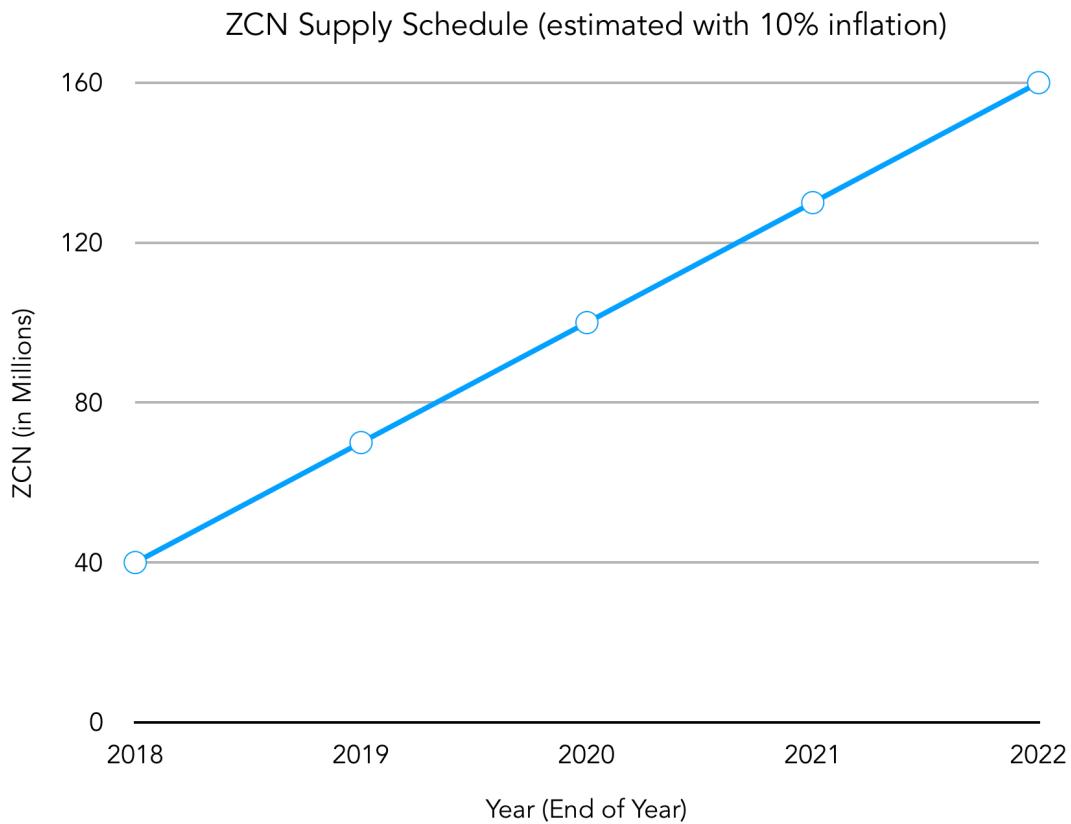


*Future Reserves unlocks only if the per unit market price of ZCN exceeds \$10.

4.2.1 Token Supply Schedule

The current inflation rate of ZCN will run at 10% per year of the outstanding token supply. The 120m tokens from quadrants “B) Team & Advisors” & “C) Seed & Community” are vested linearly over a 4 year period beginning in January 2018. To summarize, at the end of 2019 there will be 70m ZCN outstanding, 100m in 2020, 130m in 2021, 160m in 2022, etc. The 40m ZCN from quadrant “D) Future Reserves” will fully vest after four years (if, and only if, price exceeds \$10/ZCN).

The chart below depicts the aforementioned schedule:



*Future Reserves excluded. Future Reserves unlocks only if the per unit token price of ZCN exceeds \$10. This reserve unlocks in two 20m tranches in Jan 2020 and Jan 2022 if the \$10/ZCN threshold is met.

0CHAIN

5. GOVERNANCE

Upgrades in the context of a protocol are not just unavoidable, but critical. The mechanisms being used by public blockchains to initiate protocol upgrades leave a lot to be desired. Today, changes are slow, and occasionally result in a contested fork; which can have dire implications, as evidenced by the divisive chain splits seen on the Ethereum and Bitcoin networks. 0Chain governance enables a fast but fair approach to all issues, ranging from configurable changes to major code upgrades.

6. ENTERPRISE MARKET

Today's blockchain solutions for the enterprise market are disjointed. While HyperLedger, Corda, and Ethereum platforms provide a barebones blockchain (or block-less in the case of Corda), they do not solve issues regarding governance, profit sharing, addition/removal of consortium members, and verifiable storage of data. Most of the latter issues need to be developed, verified, and agreed upon by all parties so that they can trust the blockchain system. Thus, it is not simple for enterprises to transition to a blockchain system and roll-out new products on it.

In addition, the notion of a private blockchain in a datacenter under the control of a central party (or developed by such) always has the nagging complaint of being a glorified version of a traditional centralized system. And so, it is likely that a centralized blockchain system is a good transition market for enterprise products before they move to the public chain in the future. 0Chain provides a suite of protocols that addresses all of these issues, and so enterprise private chains can use 0Chain to abstract out the infrastructure and protocols, in order to develop and market new applications at a faster pace.

Section 1

An Introduction To ØChain



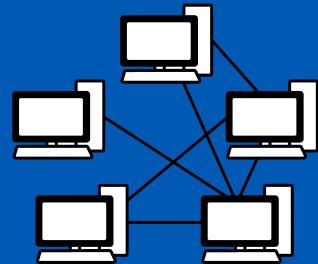
ØWallet

Secure Wallet



ØBox

Storage App



ØChainNet

Fast Blockchain Infrastructure

ØCHAIN

ØChain aims to introduce new products and services based on ØChainNet, as it conducts ongoing research, development, and code releases for ØChainNet.



ØChainNet is a fast, secure, public enterprise-grade blockchain powered by an innovative and original consensus protocol. It enables decentralized applications such as ØBox to abstract infrastructure and zero-trust protocols and build a fiat application on ØChain. ØChainNet is secured by a native cryptocurrency (ZCN), which enables a fast, secure, and free way to store and transfer value.

0CHAIN

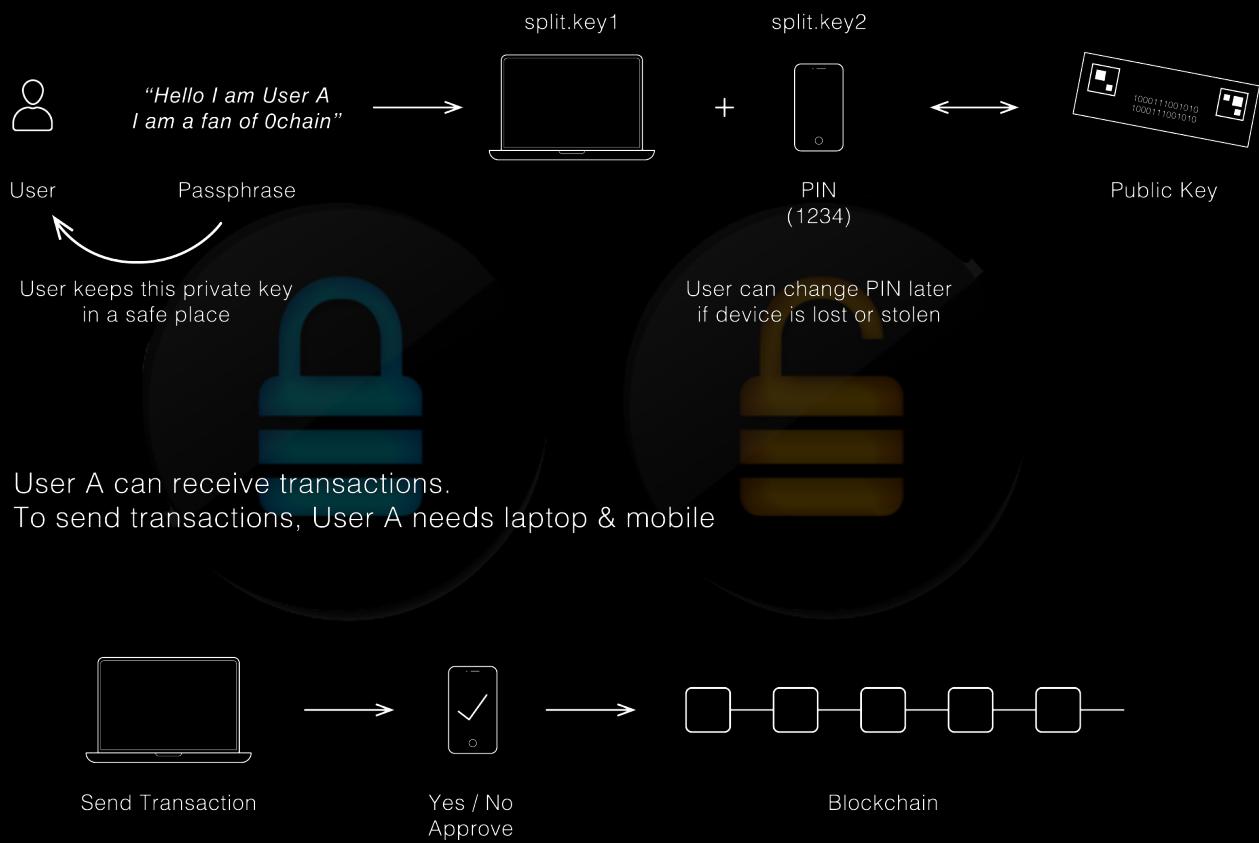
0Box: transparent, zero-trust dStorage for your data.



0Box is the first product **0Chain** will unveil on **0ChainNet**. Similar to how a browser facilitated wide adoption of Internet protocols, the intention of **0Box** is to facilitate wide adoption of public blockchain protocols. **0Box** is a platform to protect data privacy and provide transparent content monetization for both the consumer and the enterprise.

ØCHAIN

Split-Key Wallet: the convenience of a software wallet; the security of a hardware wallet.



The **Split-Key Wallet** is ØChain's breakthrough innovation for crypto wallets and passwordless login security. No more software wallet hacks. More secure than traditional 2FA, and more convenient than a hardware wallet. All you need are 2 devices and your PIN to send a transaction.

Section 2

Products

2. Products

0ChainNet is designed to serve both the crypto and enterprise industries, with the ability for enterprises and blockchains to leverage **0Box** for their decentralized storage requirement. Both industries have the ability to integrate with **0Box** and use the **0ChainNet** infrastructure as a high-performance zero-trust cloud service.

2.1 0ChainNet

2.1.1. MULTI-SIG WALLET FOR THE CORPORATE ENVIRONMENT

The 0Chain platform has a built in multi-sig wallet, and unlike other implementations, is cryptographically secure to enable m-of-n keys to execute a smart contract. Exchanges and corporations can use it to dispense funds based on several signatures, instead of one signature. It creates protection from multiple bad actors, and hacks on multiple servers. By having the multi-signature feature built into the platform, any smart contract can use this feature for a highly secure execution of value transfer.

0CHAIN

2.1.2. SPEED

The speed of a transaction is ultimately determined by when that transaction is fully processed and confirmed, also known as finality.

These are the definitions of finality:

- 1) **Steady-state block finality** is based on the rate at which a finalized block is created every block generation round, and can be directly correlated with transactions per second. So, if the block size averages 1000 transactions, and steady-state finality is a second, then the transactions per second (TPS) is 1000.
- 2) **End-to-end block finality** is defined as the time taken from the generation of a block to the finalization of that block; this timeframe is typically about two to three times that of steady-state finality.
- 3) **Transaction finality** is based on the time the transaction is sent by the user, to the miner, to the time a block with that transaction is finalized. Transaction finality is typically longer than block finality by about a 1.5 to 2 times multiple, because the transaction depends on the internet speed quality of the end user. Even if the connection is perfect, the transaction submitted could miss the block generation event. In our experiments, which included 130 world-wide nodes (100 miners, 30 sharders) with an average network latency of 700 ms, the steady-state finality was roughly ~1.2s, block finality was ~4.6s, and transaction finality was ~5.4s.
- 4) **Deterministic finality** is when you have 100% certainty that there will be no rollbacks. This lags all finality by a multiple that will be discussed in a blog in the future.

In another context, a 0Chain self-fork could have a distribution of nodes strictly within the U.S., and then the network latency would be about 200ms. Additionally, if the transactions are less than 100 per block, the steady-state finality would clock in at 300ms, with block finality at 900ms, and transaction finality at about 1s. With such speed, a decentralized exchange would be extremely fast, rivaling today's centralized exchanges and solving the problem of token custody & transparency.

2.1.3. SCALABILITY

It's expected for every chain on 0ChainNet to accommodate at least 1,000 transactions per second without a hitch. As the number of cores, memory, SSD, and IO capability increase, scaling as high as 100,000 transactions per second is possible; but this will incur high costs, and is unnecessary for any app today, including VISA. A more economical solution is scaling vertically, through additional chains based on demand. The best way to scale is to have a self-forking feature, whereby a forked chain can be formed with all associated protocols, features, and abstracted infrastructure in a trustless fashion.

2.1.4. COMPETITION

Several public blockchain solutions exist on the market today (e.g., Ethereum, Dfinity, EOS, Tezos, Tron, etc.), as well as private ones (e.g., HyperLedger and Corda), all with their own advantages. ØChain is carving out its own identity as the first enterprise-grade dStorage blockchain platform with unparalleled transaction speeds, scalability and an integrated asset-backed token economy, that can integrate with existing blockchain solutions.

2.1.5. GO-TO-MARKET STRATEGY

Building a robust community and a strong developer environment is not a simple task. One of the core strategies is to have a constant supply of bounty and competition-issued tokens to promote growth of the network. ØChain expects to be a participant in mining, and some of the accumulated interest proceeds will go toward such marketing efforts. The following strategies will be considered to educate the market:

- Bounty-issued locked and unlocked tokens for transaction and storage, based on shared link referral.
- Bounty-issued locked and unlocked referral tokens; referrals verified by the team or smart contract code.
- Bounty-issued locked and unlocked tokens for bug fixes on our governance protocol.
- A dApp “Starter Package” of locked and unlocked tokens, rewarding early applications built on the ØChainNet platform.
- Token incentives for researchers at Universities to conduct research on our protocols. Winner selected using the ØChainNet governance protocol.
- App competition. Winner package including locked and unlocked tokens, selected via the governance protocol.

As ØChain LLC raises additional capital, we expect to spend a large portion on marketing efforts in the following areas:

- University initiatives
- Meetup groups
- Accelerator engagements
- ØChain Conferences
- Hackathons

2.1.7. PARTNERS

As a small team, ØChain must be selective with our partners. We are working closely with the following impactful opportunities:

2.1.7.1. PollGateWay – eVoting/eOpinion platform

A startup in India addressing their customers' issue of transparent, authentic, and anonymous voting processes via the blockchain. However, most notably, the core issue being solved is efficiency: long lines, hours of waiting, retries, recounts, and setting up booths are all inconveniences solved by a more efficient voting process.

2.1.7.2. MyntCoinz – White labeled loyalty platform

A U.S. startup offering a BaaS (Blockchain as a Service) platform for the “*loyalty rewards*” space. MyntCoinz offers a unique solution to transform various customer engagement issues, such as a lack of flexibility and liquidity in the loyalty market.

2.1.7.3. Department of Homeland Security (DHS) – Identity fraud prevention platform

DHS seeks technical solutions that can serve the needs of U.S. Customs and Border Protection (CBP), U.S. Citizenship and Immigration Services (USCIS), and Transportation Security Administration (TSA). DHS is interested in using blockchain to address the challenges of interoperable digital entitlement attestations that support individual control and accountability of data release, while incorporating digital counter-fraud technologies and tactics, enterprise lifecycle management, and a high degree of usability across service delivery modalities.

2.1.7.4 AWS – Infrastructure and BaaS partner

We have been working with AWS for over 6 months and have been granted \$100k of cloud resources, which has been valuable in rolling out our Blockchain as a Service platform. We also passed their technical requirement for disaster recovery and cloud service processes to get accepted in their Advanced Solution program. Moving forward, we'll work with their sales team on opportunities.

2.1.7.5 Oracle – Infrastructure and BaaS partner

We are a partner in Oracle's standard program, and plan to bring our BaaS to their customers. One of their requirements to do so is to have a platform up on their datacenter. Given our limited resources, we have decided to wait until the AWS opportunity and ØChainNet are both live in the marketplace — something we expect for Q2 2019.

2.2. 0BOX

The second product that we're developing in parallel is 0Box, a data privacy and transparent decentralized storage (dStorage) platform. 0Box will be used to accomplish two core objectives:

1. Showcase a native dApp that runs on 0ChainNet as a model for other dApps.
2. Facilitate use of decentralized storage on 0ChainNet to “seed” adoption of blockchain.



2.2.1. HOW IT WORKS

Users lock tokens or directly allocate required tokens for a specific storage **allocation** (e.g. 1TB). The allocation is created one time when the miner randomly chooses the least expensive blobbers. The list of blobbers are on the blockchain, since each service provider of 0Chain needs to register with the network. If the blobbers offer the same price, then they are randomly picked. The client collects all signed contracts from the blobbers and sends a transaction with the list to the blockchain. Then it starts uploading files to the blobbers. Once the file is committed, people can read from the blobbers. The health (reliability, availability) of the file is checked on a regular basis through challenges, and if anything is amiss, then the repair protocol takes care of it in the background.

Users can upload as many files as they wish. Files can be private or public and can be shared via links on any platform. When a user clicks on the link, it invokes the 0Chain mobile or desktop app to download the file.

2.2. **0BOX** (cont'd)

Regarding competition, we do not directly compete with DropBox, Box, or other data sync solutions, since our target market are users that value data privacy and transparency — for which a practical solution does not exist. These consumers care about their personal data, especially their images, posts, and videos. 0Box enables them to protect their images and videos by placing them on 0Box, and pasting a link on their social wall for their friends to download and view those images and videos. Users will be able to monitor a file's downloads on the blockchain, and be assured of their privacy.

Consumer content platforms such as YouTube, Spotify, Apple, and Pandora do not address small artists. New rules alienate small publishers. 0Box can accommodate both of the following types of content creators:

- 1) A well-known publisher, with a large social media following (1m followers or more)
- 2) A small publisher, with a small social media following (1k followers or less)

The small publisher can host content on YouTube and share the video for free, or host it on 0Box (for free via locked tokens) and charge a nominal fee to view their content. In the latter case, the user generates some revenue instead of none. If the well-known artist has a million followers, they can share the 0Box link and price it lower than they expect to get from ad clicks. The artist then promotes the content on social networks for people to watch it. The average cost per thousand impressions (CPM) for YouTube is \$9.68 and \$3.21 for CPC (2018). A video with 1000 views would earn them 9.68. If the artist sells the video for \$0.25c to their fans, then only 40 out of 1000 followers need to download it to achieve the same result, but the artist gets to keep all of the revenue. For a user's perspective, they can get access to the content, free of ads, with their interest tokens, and still directly support the artist. 0Box provides a win-win solution for content creators and consumers. The monetization feature of 0Box will be offered later in the roadmap.

2.2. **0BOX** (cont'd)

For bloggers and other content creators, they can utilize 0Box share links for monetized content, and get paid via a share link. They can simply write a teaser article on Medium, Twitter, Facebook, or brand name journal publication, and have people download the full content via the share link.

2.2.2. COMPETITION

While several providers of storage and data sync solutions exist in today's market, 0Box is addressing an unserved need for data privacy, security, and a transparent dStorage.

2.2.3. GO-TO-MARKET STRATEGY

Similar to the 0ChainNet strategy, the 0Box viral marketing strategy for building out a strong network will be accomplished through a combination of the following:

- Bounty-issued (locked and unlocked) ZCN tokens for creative artists.
- Bounty-issued (locked and unlocked) ZCN tokens for 0Box based on referrals.
- Reward-issued (locked and unlocked) ZCN tokens for content competitions (eg, “best creative content” contest, with winner determined via the 0Chain governance protocol)
- Bounty-issued (locked and unlocked) ZCN tokens based on most downloaded content.

2.2.2. GO-TO-MARKET STRATEGY (*CONT'D*)

We also expect to allocate substantial funding towards marketing efforts in the following areas:

- Specific Content groups (art, graphics, music, video, clips, gifs, blogs, reports)
- Privacy meetup groups
- Accelerator engagements using API using 0Box
- 0Chain Conferences

2.2.3. TELECOM CHANNEL PARTNERS

We are working with three telecom operators to leverage our 0Box solution to provide a better cloud service for their customers, and increase their top line with higher data rates. Our proposal is a freemium data model for bandwidth related to cloud usage. In doing so, subscriber acquisition cost reduces substantially, increasing their top line over time with specific data plan introductions built around 0Box.

2.2.4. CYBERSECURITY CHANNEL PARTNERS

We are selectively pursuing customers and partners that can most effectively leverage 0Box to address the issues of data privacy and data breach, particularly within the healthcare sector. With 0Box's hassle-free integration platform for existing websites, internet processes, and healthcare companies, a 0Box solution is practical for both the consumer and enterprise.

2.3. 0WALLET

Our 0Wallet product allows for secure value transfer without the need for hardware wallet technology. A 0Wallet user can keep their secret phrase in a safe place, and use it to generate “split-keys”, which are factor authenticators via a device of the user’s choosing, such as a laptop and a mobile phone. Since the pin code is used on the second device, even if one of the devices or both devices are compromised, the user needs the pin code to send a transaction. There is no need to remember back-up codes for traditional 2FA, nor a need for hardware wallets.

Split-key wallet technology can be used for different use cases like password-less logins for websites and applications. It can be also be used with hardware devices such as RSA type keys used in banks. Additional use cases are keyless entry for cars and buildings.

For more details, reference Appendix 5 in the Appendices section.

Section 3

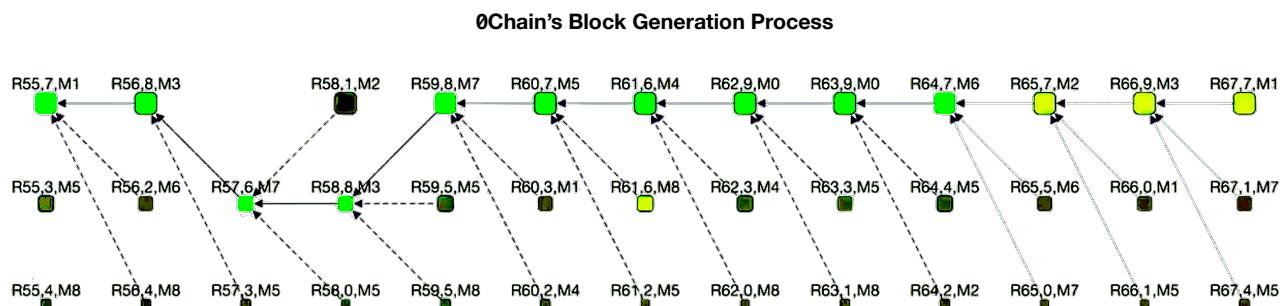
ØChain Architecture

3. 0Chain Architecture

The value of 0ChainNet is tied to the protocol design and implementation. We believe we have an unrivaled, innovative design not seen in today's Distributed Ledger Technology (DLT) space, especially with regards to network security and asset-based service solutions. Many of these protocols have been rigorously tested and demonstrated on private and public test networks.

3.1. CONSENSUS PROTOCOL

0ChainNet offers a fast, secure, and scalable blockchain through a proof-of-stake consensus protocol that extends existing work of Dfinity protocol in several ways. The 0Chain protocol assigns various parties in the system with specialized roles: 0Chain has multiple generators produce blocks to prevent DDoS attacks, generate random numbers, and verify blocks; sharders store the blockchain history and respond to queries about that history; and blobbers store data needed for dApps.



This design allows for more specialized machines to be used for each of these roles. Otherwise queries will bog down miners, and storing large data files will slow down miners, if they were to do all the tasks. 0Chain also introduces a squared staking approach for Sybil resistance, by which miners and sharders are probabilistically chosen based on the square of the number of tokens they have staked; this design incentivizes miners and sharders to stake their coins in a single account, and thus risk greater penalties should they fail to perform their duty.

Finally, the 0Chain consensus protocol makes very mild assumptions about the network latency to allow for faster confirmation time, because nodes do not need to wait a fixed time in order to progress, but can instead progress shortly after they receive their expected messages.

3.2. STORAGE PROTOCOL ¹

A key distinction of our data storage system from other blockchain storage solutions is that we divorce the role of mining from that of providing storage. Computers that provide storage are referred to as blobbers. Blobbers are neither responsible nor required for mining. In this manner, we lighten the load on our mining network and enable fast transactions on a lightweight blockchain. As the client and blobber interact, the client generates special signed receipts called markers. These markers act like checks that the blobber can later cash in with the blockchain.

Once the interaction between client and blobber has concluded, the blobber writes an additional transaction to the blockchain, which redeems the markers for 0Chain tokens and commits the blobber to a Merkle root matching the data stored. The leaves of the Merkle tree must match markers sent from the client, preventing either the client or the blobber from defrauding each other.

After a file has been stored, a challenge protocol ensures both that the blobber continues to store the file and continues to be paid for that work. The mining network posts a transaction, challenging the blobber to prove that it still possesses the data that it was paid to store. The blobber must provide that data, the relevant system metadata, and the client-signed marker to prove that the right data is stored. The blobber is then rewarded or punished accordingly.

With our design, the majority of the work between clients and blobbers happens off-chain. The mining network is only involved enough to ensure that clients pay blobbers for their work and that the blobbers are doing the work that they have been paid to do. Our design assumes that the client is using erasure codes to ensure greater resiliency. While this is not a strict requirement, it does enable a client to recover if a blobber proves to be unreliable.

¹ Will be presented and published at IEEE Dappcon in April 2018. Early publication at <https://0Chain.net/research>

3.3. SPLIT-KEY WALLET PROTOCOL ²

The split-key wallet protocol uses a BLS signature scheme to split keys and let users interact using crypto keys via a blockchain. Since the cryptocurrency balance is maintained against these keys, it's very important to protect the private key. The private key is split into two secondary keys, storing each of the secondary key on a different device. Signing requires individual signatures from each device. Hence, losing any one device can still protect the primary key. In addition, if desired, one of the secondary keys can be further split into two parts; only one of which is stored on the device and the other is a simple PIN that the user has to enter each time. This provides an extra layer of protection in case both devices are compromised. The split-key wallet protocol makes it easy to generate as many split keys as desired providing the ability for the user to periodically rotate the split keys and in the process change the PIN.

3.4. TOKEN REWARD PROTOCOL ³

When clients lock tokens, they are rewarded with an “interest”. The interest is newly generated ZCN tokens, intended (but not required) for payment of services on the network. These services can be miner compensation for transaction processing, blobber compensation for storage, or transmitted to any other client in exchange for a service; facilitating a lucrative market for building and running distributed applications. In the event of network congestion, a client may also offer to lock a greater amount of tokens to ensure that their transaction is accepted by the mining network. The token reward protocol creates an economy where ZCN tokens can be used to receive services for “*free*” — meaning, the client does not lose their initial stake, but still adequately compensates the service provider.

² Will be presented and published at IEEE Dappcon in April 2018. Early publication at <https://0Chain.net/research>

³ Will be presented and published at IEEE Dappcon in April 2018. Early publication at <https://0Chain.net/research>

3.5. GOVERNANCE PROTOCOL

Our governance protocol enables simple implementation of a variety of lightweight, non-controversial changes, while still supporting more extensive and potentially controversial changes. Our protocol provides this flexibility by supporting different thresholds for different types of changes, which we divide into configuration changes, moderation, and feature requests. We expect that moderation will be relatively non-controversial, that feature requests are likely to be highly controversial, and that configuration changes might be either.

Our design builds on our token-locking reward model. Essentially, clients who own 0Chain tokens may temporarily lock tokens to produce token rewards for service providers. In our voting protocol, these token rewards are treated as votes; by locking more tokens, clients may dedicate more votes to a proposal that they favor. Similarly, they may allocate token rewards against any proposals that they oppose. A critical aspect of our design is that our voting mechanism can measure both whether a proposal has broad community support and the degree of support or opposition from different parties for a specific proposal.

One concern in voting protocols is that a wealthy supporter of a proposal could sweep in during the last moment of voting to pass or defeat a measure before other members of the community can react. This is a particular concern for our protocol, since clients have an economic cost associated with voting for a proposal. Our design addresses this issue by having multiple rounds of voting. If a proposal passes, it is followed by a review period where the community may veto the proposal. If a proposal is vetoed, the community may vote to override the veto. The override itself may be vetoed, which may also be overridden, and so on. Eventually, one side or the other will exceed a threshold that the other side cannot match, and the issue will be settled. To minimize the back and forth votes, the vetoing/overriding faction must exceed the threshold by a fixed buffer amount.

3.5. OTHER PROTOCOLS

*Additional protocols are in the works related to View Change, Self-Forking, Proxy-Reencryption, and Economic protocols, as well as The Equilibrium Price of ZCN, and The Mathematical Valuation of ZCN Token. The overarching objective with 0Chain's library of protocols is to provide a complete "a-to-z" platform for a dApp, such as 0Box, to seamlessly operate on our chain without the need to worry about infrastructure and protocols.

Section 4

Appendices

Appendix 1: Team

The inception of **0Chain** began in July 2017, which led to a white paper in December 2017 and funding in February 2018. We have a world-class team, headquartered in downtown San Jose, heart of Silicon Valley. We think out of the box and are impassioned by blockchain technology.

For more details visit <https://0chain.net/team>.



SASWATA BASU
CEO & FOUNDER

Our founder and CEO, Saswata Basu, has 20 years of experience with startups and corporate product development. His first startup, InSpan, had a successful exit in 2000 to CommScope. He has worked on startup initiatives at Intel and Harris in mobile, wireless backhaul, IoT and energy efficiency sectors.



TOM AUSTIN
CO-FOUNDER

Our co-founder, Tom Austin, is an Associate Professor at San Jose State University, and is a well-renowned cyber security and programming language expert. He is the inventor of our storage and token reward protocols.



SIVA DIRISALA
CTO & VP OF ENGINEERING

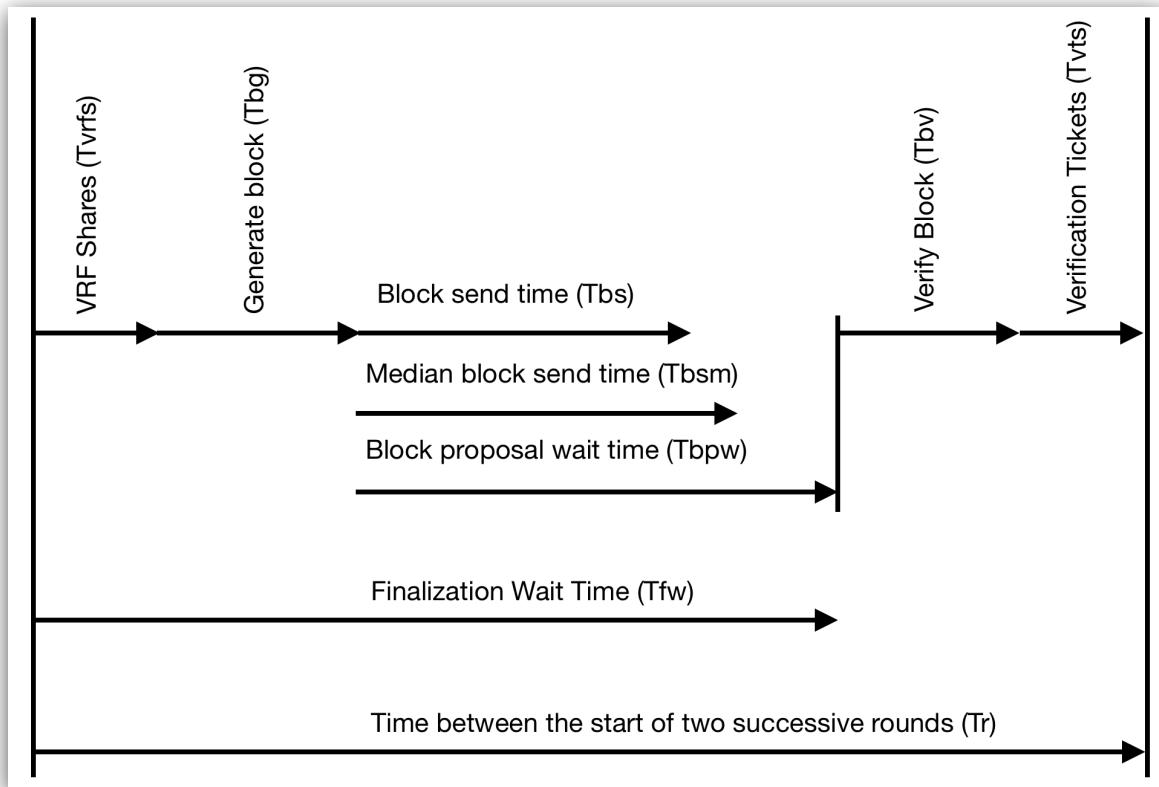
Our Chief Technical Officer, Siva Dirisala, has 20 years of experience as a computer architect and product developer at Service Now & Oracle, and has been solving complex algorithmic and implementation problems his entire career.

Appendix 2: Understanding ØChain Finality

The ØChain consensus protocol has been tested on a large cluster of 100 miners with 4 block generators per round and 30 sharders with 6 block replicators per round. The testing is done in a non-Byzantine condition but realistic network conditions by spreading the 100 miners across 14 different data center zones spanning the world (California, Canada, Frankfurt, Ireland, London, Mumbai, Ohio, Oregon, Paris, Seoul, Singapore, Sydney, Tokyo and Virginia).

These experiments and associated data points provide empirical evidence to how the finality time is related to the various underlying parameters. The following diagram illustrates a full cycle of what happens between two successive rounds getting started.

1. CONDITIONS



When a notarized block is first discovered in a round, a miner does two things:

- 1) Start the next round.
- 2) Wait to start the finalization process for the previous round.

These two processes are independent and are shown as two separate timelines above. We define the steady state finality of a block is the time between invoking finalization between two successive rounds. So, as long as the finalization wait time, T_{fw} , is less than the total time to generate, send and verify a block referred to as round processing time, T_r , then the steady state finality of a block is given by T_r .

1. CONDITIONS (CONT'D)

```
Tr = VRF Share send time + Block Generation time + Block Proposal Wait time + Block  
Verification Time + Verification tickets send time.
```

Note that this is the ideal time where all of these are assumed to happen simultaneously at all the miners, but in reality there will be some variance of when each of these steps are started at each miner.

The above formula gives the following key insights

- 1) If the block proposal wait time is sufficiently large, every node receives all the blocks being proposed for the round before the verification starts. This ensures only one block is signed (for honest nodes)
- 2) If the block proposal wait time is too little, most blocks will arrive after the wait time and hence the chance of signing lower ranked blocks increases resulting in several blocks getting notarized in a given round.

Another option is to use a dynamic block proposal wait time. Under this scheme, as long as there is at least one generator whose network time is less than the median network time, a miner will choose that as the block proposal wait time. If in a round, there are no generators whose network time is less than the median, then the configured block proposal wait time will be used. If there are multiple generators whose network time is less than the median, then among all these generators whoever has the higher ranked block, their network time is selected as the block proposal wait time. This scheme tries to balance between waiting arbitrarily too long and too short striving to speed up the finality and reduce number of notarized blocks in a given round by signing fewer blocks.

1. CONDITIONS (CONT'D)

We define steady state finality (time between running two successive finalizations) and start-to-finish block finality (time between the starting of a block and eventually finalizing it). Note that the finality here is a local view of the individual miner.

- Steady state block finality time = T_r (when $T_{fw} < T_r$)
- Start-to-finish block finality time = Finality Lag * $T_r + T_{fw}$
- Where finality lag is the number of blocks by which the finality lags.

2. RESULTS

In our experiments, we found that the block generation time and verification times are significant and comparable to network times and even the network time itself is different for short messages such as sending VRF shares and verification tickets vs sending a large message like a block. For example, the block message time was 3 times or more than the short message time. Based on these observations, in our protocol, every miner maintains the time it takes to send a small message and a large message to every other miner and can be configured to use these data points to dynamically optimize the finality time.

2. RESULTS (CONT'D)

In our experiments, we found that the block generation time and verification times are significant and comparable to network times and even the network time itself is different for short messages such as sending VRF shares and verification tickets vs sending a large message like a block. For example, the block message time was 3 times or more than the short message time. Based on these observations, in our protocol, every miner maintains the time it takes to send a small message and a large message to every other miner and can be configured to use these data points to dynamically optimize the finality time.

The experiments were conducted by varying the block proposal wait time, wait mode (static vs dynamic) and the finality wait time.

The below table shows the steady state block finality, start to finish block finality and percent of rounds notarized with 1, 2, 3 and 4 blocks (as we have 4 generators). The Tfw, the finalization wait time, is twice the Network Relay time indicated in the table. In our experiments, the finality lag is 3 blocks. Hence, Start-to-finish finality time = 3 * Tr + Tfw.

Network Relay (ms)	Block Proposal Wait (ms)	Block Proposal Wait Mode	Steady state block Finality	Block Start To Finish	Rounds	Blocks Notarized = 1	Blocks Notarized = 2	Blocks Notarized = 3	Blocks Notarized = 4
200	200	Static	1238.57 ±143.22 ms	4497.22 ±362.72 ms	25805	24539 (95.09%)	1262 (4.89%)	5 (0.01%)	0 (0.00%)
600	200	Static	1251.13 ±159.85 ms	5338.80 ±365.94 ms	28762	27363 (95.13%)	1391 (4.83%)	9 (0.03%)	0 (0.00%)
400	400	Static	1267.28 ±140.92 ms	5013.01 ±357.77 ms	27842	26525 (95.26%)	1318 (4.73%)	0 (0.00%)	0 (0.00%)
200	800	Static	1424.32 ±119.25 ms	5046.32 ±330.31 ms	25598	25454 (99.43%)	145 (0.56%)	0 (0.00%)	0 (0.00%)
200	200	Dynamic	1261.21 ±150.29 ms	4576.53 ±376.98 ms	25988	24804 (95.44%)	1178 (4.53%)	7 (0.02%)	0 (0.00%)
400	400	Dynamic	1262.96 ±153.61 ms	4995.64 ±365.48 ms	26189	24966 (95.32%)	1220 (4.65%)	4 (0.01%)	0 (0.00%)
200	800	Dynamic	1260.01 ±142.89 ms	4564.89 ±359.27 ms	26149	25088 (95.93%)	1059 (4.04%)	3 (0.01%)	0 (0.00%)
200	1600	Dynamic	1306.84 ±186.77 ms	4710.29 ±409.20 ms	25726	24963 (97.03%)	763 (2.96%)	1 (0.00%)	0 (0.00%)

3. OBSERVATIONS

The following observations have been made based on the aforementioned results:

1. No rounds with 4-notarized blocks were observed in any of the scenarios.
2. With static block proposal wait time, the steady state finality increases little from 1238.57 ms to 1267.28 ms when the wait time increases from 200 to 400 but increases much higher to 1424.32 when the wait time is 800. When the wait time is too short, it is still required to wait to receive the first block. So, there is a minimum built-in wait time. Hence, any block proposal wait time below this minimum wait time will not reduce the steady state finality. Similarly, any wait above the time when all blocks are received will result in delaying the verification process and hence directly contribute to the round processing time, T_r .
3. As the static wait time increased, the percent of rounds with single notarized blocks increased showing very high percent at 800 ms wait confirming that majority of the nodes are able to send a block within this time.
4. Increase of the finality wait time (T_{fw}) from 400 to 1200 didn't impact the steady state finality much by only increasing it from 1238.57 to 1251.13. This is expected because, as long as the finality wait time is less than, T_r , the time between two successive rounds, the steady state finality only depends on T_r . Since, T_r is 1238.57 when the finality time was 400, increasing it to 1200 didn't significantly increase the steady state finality.
5. However, increase in T_{fw} , does increase the start to finish block finality linearly. For example, the finality increased from 4497.22ms to 5338.80ms and $4497.22+3*(1251.13-1238.57)+(1200-400) = 5334.90$ which is close to the observed value of 5338.80 and agrees with the start to finish finality time formula given above.
6. While static wait of 800ms resulted in 99.43% rounds with single notarized blocks, 800ms dynamic wait resulted in only 95.93% but steady state finality reduced from 1424.32ms to 1260.01ms. This is the expected trade-off between improving finality and ending up with multiple notarized blocks in a round.
7. Unlike the static wait, steady state finality is not impacted much with increasing the block wait proposal time under dynamic wait. This is again expected because most of the time the wait will be within the median network time and only occasionally it will exceed that but still be the network time of one of the generators that is smaller than the dynamic wait time. Hence, even after increasing the dynamic wait time to 1600ms, the steady state finality hardly had any impact.

These experimental results provide valuable insights for us to fine tune the key parameters as needed. In dFinity, the process timings such as block generation and verification times are assumed to be 0. The network time is considered the same for a small message and a large message. In ØChain protocol implementation, the process timings are explicitly considered and also the time for small and large messages is treated separately.

Appendix 3: Consensus Protocol

The 0Chain Consensus Protocol

Jonathan Katz^{1*}, Thomas Austin², Siva Dirisala³, and Saswata Basu³

¹ Dept. of Computer Science, University of Maryland.

² 0Chain LLC and San Jose State University.

³ 0Chain LLC.

Abstract. We describe the 0Chain blockchain ecosystem, including a new consensus protocol offering fast finality. We provide proofs of security for the protocol, along with experiment results validating its efficiency under realistic network conditions.

1 Introduction

Since the advent of Bitcoin [Nak09], the blockchain has revolutionized the world of cryptocurrencies and distributed computation. Ethereum [Woo14] further developed this promise by integrating Turing-complete smart contracts into the blockchain for building distributed applications (dApps).

Despite the promise of blockchain protocols, they have been held back by their slow consensus times. For example, in Bitcoin a transaction is not considered finalized until it is six blocks deep in the chain, a process which takes roughly one hour. Newer protocols have attempted to address this limitation by introducing consensus algorithms with faster finalization times.

One such protocol, Dfinity [HMW18], uses a randomness beacon (implemented via a *verifiable random function*, or VRF) for ranking different proposed blocks. The designers also introduce the concept of *notaries* who sign the highest-ranked block in each round. The authors describe notarization as “optimistic consensus”; in most rounds, only one block will be notarized, and in that case the unique notarized block will be finalized soon thereafter. Importantly, only notarized blocks will be accepted as part of a chain by miners; this prevents both selfish mining [ES18] and the “nothing-at-stake” problem [Poe15].

0Chain offers a “fast, flexible, free” platform for dApp development through a proof-of-stake consensus protocol that extends previous work in several ways. First, the 0chain protocol assigns various parties in the system specialized roles: at any given time, a subset of the clients (referred to as the “active set”) serve as *miners* running the consensus protocol; in turn, a subset of the miners act as *generators* proposing new transactions. *Sharders* store the blockchain history and respond to queries about that history; and *blobbers* store data needed for dApps. This design allows for more specialized machines to be used for each of these roles; by reducing the number of parties running the consensus protocol at any point in time, it also reduces network latency thus improving finalization

* Work done as part of a consultancy agreement with 0Chain LLC.

Appendix 4: Storage & Token Reward Protocol

Lock and Load: A Model for Free Blockchain Transactions through Token Locking

Paul Merrill and Thomas H. Austin

*Research Group / Department of Computer Science
0Chain LLC / San José State University
San Jose, United States*

Jenil Thakker and Younghée Park

*Department of Computer Engineering
San José State University
San Jose, United States*

Justin Rietz

*Department of Economics
San José State University
San Jose, United States*

Abstract—Bitcoin introduced the world to blockchain-based cryptocurrencies, and Ethereum highlighted their value in building distributed applications (dApps). However, the development of blockchain-based applications has been held back by high transaction fees.

In this paper, we introduce a model for free transactions on the blockchain. Rather than spending tokens for transaction fees, a token owner (known as a *client*) locks tokens to generate new tokens as a reward for the miner who includes the transaction in a block. This *token-locking reward model* eases congestion on the blockchain in the same manner as fees do in protocols like Bitcoin and Ethereum, but without forcing clients to sacrifice their tokens.

This same design can be used to incentivize service providers. We show how a client can lock their tokens to generate new tokens for storage providers, and how this reward mechanism can help to facilitate an audit of the storage provider.

Index Terms—blockchain, storage, cryptocurrency economics

I. INTRODUCTION

Our blockchain introduces a *token-locking reward model*; rather than spending tokens, clients lock tokens to pay for services. This act creates more tokens, which can be used to reward miners or other entities for their work.

This model is similar to Bitcoin’s design [1]. Early in Bitcoin’s history, miners were primarily incentivized to generate blocks through *coinbase transactions* that rewarded miners with newly created bitcoins. As interest in Bitcoin has skyrocketed, clients must offer transaction fees to motivate miners to include their transactions. This design also serves to ease congestion; when demand for transactions increases, clients can raise these fees to increase the odds of their transactions being accepted.

Our model can be seen as a blend of these two mechanisms. Clients lock tokens to generate new tokens for miners, similar to coinbase transaction rewards; but as with transaction fees, a client may offer to lock a greater amount of tokens to ensure that their transaction is accepted by the mining network.

When clients lock tokens, they can give the newly generated tokens to any other client, facilitating a market for creating distributed applications (dApps). In this manner, tokens in our network can be used to buy services for “free”, in the sense that the client does not lose their tokens, but still gives something of value to the service providers.

Using our token-locking reward model, we build a sample dApp for storage, the blockchain-observable storage system

(BOSS). With BOSS, storage providers are rewarded for their work by clients who lock tokens. We also show how BOSS can ensure that neither clients nor storage providers can cheat one another; this process relies on special, signed *markers* that ensure public agreement between the two parties. Additionally, this agreement can be publicly validated; third parties can verify that the storage provider is storing the agreed-upon data using nothing more than public transactions on the blockchain and signed messages from the client.

A key property of our design is that the clients may give themselves the newly generated tokens (hereafter referred to as *interest*). An alternate design could restrict a client to only reward service providers. However, that approach would incentivize clients to feign services in order to mint new tokens; we legitimize this behavior and eliminate such shenanigans.

An interesting economic consequence of this design is that it reduces the opportunity cost of holding a token versus holding a fiat currency in an interest bearing bank account. The interest paid at least partially offsets a possible reduction in the token value. If the level of interest paid moved inversely with the token price the interest payment might substantially offset changes in token value, which could be a stabilizing factor. If the token price decreases, people will lock more tokens in expectation of receiving a higher interest rate, and this locking of tokens in effect reduces supply, creating upwards pressure on the token price.

Our paper makes the following contributions:

- We present our token-locking reward model, which enables clients to reward service providers by locking tokens, without needing to sacrifice their tokens.
- We demonstrate how our model can be used to incentivize miners to accept transactions and generate blocks.
- We use our token-locking reward model to build a storage dApp, allowing clients and storage providers to negotiate an agreement for service. As with incentivizing miners, we can reward storage providers for their work without requiring the client to sacrifice tokens.
- We show how signed markers and the blockchain can be used to validate a storage provider’s work.
- We provide an economic analysis of our system, showing how our model can reduce the price instability typically associated with cryptocurrencies.

Appendix 5: Split-Key Protocol

Splitting and Aggregating Signatures in Cryptocurrency Protocols

S. Sharmila Deva Selvi¹, Arinjita Paul¹, C. Pandu Rangan¹, Siva Dirisala², and Saswata Basu²

¹Department of Computer Science and Engineering, IIT Madras, India
Email: {sharmila, arinjita, prangan}@cse.iitm.ac.in

²0chain LLC, San Jose, USA
Email: {siva, saswata}@0chain.net

ABSTRACT

The blockchain technology and a vast amount of cryptocurrency related activities have generated an unprecedented level of interest among the public. However, even at the entry level, cryptocurrency users need to deal with the complex task of key management. In this paper, we propose a simple way to manage a user's private key, under a reasonable assumption that the user has two devices at his disposal (say a laptop and a mobile phone). We refer to our strategy as *key splitting*. Since these cryptographic keys are used for generating digital signatures, we should take a closer look at the signature schemes that would perform best under key splitting. At the operational level, scalability is one of the main challenges faced by the users and developers. While there are fundamental issues like consensus that challenge scalability, we focus on the computational efficiency in a block formation. Aggregation of signatures is one of the effective solutions to this problem. To this end, we observe that none of the existing signature schemes work well for BOTH key splitting and aggregation. The current popular schemes such as the ones used in Bitcoin or Schnorr's scheme implemented over Elliptic curves are neither suitable for aggregation nor can their keys be split in a convenient and meaningful way. A detailed theoretical and empirical analysis shows that the BLS short signature scheme is best suited for achieving both key splitting and aggregation.

Index Terms—Blockchain, key management, wallet, signature, scalability.

I. INTRODUCTION

The real-world as well as the academic studies on cryptocurrencies and the block chain technology are among the most significant and trendy developments of Information Technology. Block-chain technology is witnessing an exponential growth in interest and technical advancement at this point of time. While these areas are witnessing an unprecedented growth and attention, their deployments face major hurdles at several fronts. One of the major concerns related to this technology is scalability and in general efficiency/reliability of the whole operation. For instance, every user in this community, sooner or later, directly or indirectly, is forced

to deal with challenges of maintaining and managing the cryptographic keys that are used. The subtleties and challenges involved in key generation, maintenance and management are well known in security industry and both cryptographic and policy based solutions have been devised in the past. However, in the context of cryptocurrencies, we still do not have satisfactory solutions that would help scalability or ease of use. The second major concern is related to computational efficiency of the tasks performed during the execution of the protocols. One of the most computationally intense and most frequently used cryptographic primitives in blockchain technology is digital signatures. The users need to generate every transaction with appropriate authentication done on the transaction and the minors or validators need to verify/validate the same multiple number of times. In this paper, we focus on the signing process at the users end and verification process at the block formation/validation end.

In order to handle the challenges and complexities of key management, a number of techniques were proposed and deployed in different cryptocurrencies. In Bitcoin core, the keys are maintained in local storage. A typical user will have an access to a wallet software of his choice and use the same to authenticate transactions he is generating. As wallets generate the digital signature, it requires an access to the private key of the user. While this speeds up the wallet operations, the presence of a key for a long time in a system that is online increases its vulnerability. Off-line storage and air gapped storages are used by systems such as Armory [1]. Password protected wallets are deployed by certain systems but they do not provide any security against a malware that might read the key strokes etc. Third party hosted wallets are also suggested to remove the pains of key management to a novice user but then it requires enormous amount of trust in a third party. A detailed analysis on various techniques that are currently used in practice together with limitations in their usability is reported in [7].

In view of the shortcomings of the existing systems, we take a fresh look at key generation and management using two systems that may be available with a typical user. Our proposal is simple, easy to implement, secure and offers protections against theft/loss of the systems. Given that a typical user may have at his disposal several devices(atleast two, say a laptop



1World: a decentralized ecosystem for online engagement and services

POSITION PAPER

1World Online Inc.
October 2017

Legal Disclaimer

The purpose of this White Paper is to present 1World Online and 1WO Token to potential token holders in connection with the proposed ICO. The information set forth below may not be exhaustive and does not imply any elements of a contractual relationship. Its sole purpose is to provide relevant and reasonable information to potential token holders in order for them to determine whether to undertake a thorough analysis of the company with the intent of acquiring 1WO Tokens.

Nothing in this White Paper shall be deemed to constitute a prospectus of any sort or a solicitation for investment, nor does it in any way pertain to an offering or a solicitation of an offer to buy any securities in any jurisdiction. This document is not composed in accordance with, and is not subject to, laws or regulations of any jurisdiction, which are designed to protect investors.

The product token is not a digital currency, security, commodity, or any other kind of financial instrument and has not been registered under the Securities Act, the securities laws of any state of the United States or the securities laws of any other country, including the securities laws of any jurisdiction in which a potential token holder is a resident.

1WO Token cannot be used for any purposes other than as provided in this White Paper, including but not limited to any investment, speculative or other financial purposes.

1WO Token confers no other rights in any form, including but not limited to any ownership, distribution (including, but not limited to, profit), redemption, liquidation, property (including all forms of intellectual property), or other financial or legal rights, other than those specifically set forth below.

Certain statements, estimates and financial information contained herein constitute forward-looking statements or information. Such forward-looking statements or information involve known and unknown risks and uncertainties, which may cause actual events or results to differ materially from the estimates or the results implied or expressed in such forward-looking statements.

This English language White Paper is the primary official source of information about the 1WO Token. The information contained herein may from time to time be translated into other languages or used in the course of written or verbal communications with existing and prospective customers, partners etc. In the course of such translation or communication some of the information contained herein may be lost, corrupted, or misrepresented. The accuracy of such alternative communications cannot be guaranteed. In the event of any conflicts or inconsistencies between such translations and communications and this official English language White Paper, the provisions of this English language original document shall prevail.

Abstract

1World Interactive Platform for publishers and brands combines a best-in-industry set of engagement and monetization tools (Polls, Quizzes, Debates, Trivia, Insights, Interactive Maps etc.) on the front-end and state-of-the-art Analytics / Audience Insights and various plug-ins, including A.I. powered on the backend.

1World aggregates programmatic and direct deal advertising services linked to 1World widgets and is in Pilot with additional monetization services such as **commercial data collection** in a format of paid Mini-Surveys via an existing research marketplace, traffic generation tools, and more considered in the Roadmap.

1World will introduce its own cryptocurrency (1WO** Coins or **Tokens**)** in the Fall of 2017 to increase engagement / gamification / monetization options that results in unprecedented Win-Win-Win model between Publishers, their Audiences and Services offered to them (Ads, Research, Content Syndication etc.).

1World serves as an intelligent layer connecting Blockchain-authenticated users with various services offered via smart contracts and offers business model via commission on such services delivered via 1World widgets.

*No more conflict between good User Experience and Monetization
via Ads on your site!*



Table of Contents

Legal Disclaimer	0
Abstract	2
Table of Contents	3
Introduction	6
1World Vision & Execution	9
History of 1World Development and Deployments	11
1World Today	13
Market size and opportunity	14
1World Intellectual Property (IP) Portfolio	15
1World Cryptocurrency & Use Cases	18
Readers' Use Cases	18
Publisher's Use Cases	20
Advertiser's Use Cases	20
Why Blockchain?	22
The 1World Economy	24
World Points System Today	25
1World Wallet	25
Expansion with More Use Cases	25
1World Token Issuance	26
1World Token Allocations	27
Escrow	27
Supply Schedule	28
Payments Processing	28
Token Distribution Event	29
Token Distribution for the Team	30
Use of the Proceeds	30
Technical implementation	31
Implementation Based on Ethereum ERC20	32
Current Ethereum Limitations	32
Raiden Network	34
Plasma	34

State Channels	35
Exonum	35
Smart Contracts	36
Current Implementation Plan	36
Use Cases for 1WO Tokens	36
Reader (end-users) micro-transactions to earn 1W points / 1WO Tokens	37
Reader micro-transactions to spend 1WO Tokens	38
Brands or Research related transactions	38
Publishers transactions	39
1World Reward Engine	39
1World Campaigns Management	39
Identity Management & DID	39
Summary	40
Roadmap	40
1World Technology Advisors	42
1World Blockchain Advisory Team	42
US Business Advisors	42
Silicon Valley	43
New York Media Companies	43
Global Partners	44
Industry Partners	44
1World Team	45
1World Leadership Team	45
1World US Business Team	45
1World Global Business Team	46
1World Engineering Core Team	46
1World Support Team	47
1World Publishers and Partners	48
Publishers using 1World	48
Advertisers using 1World	48
Partners integrating 1World	48
Risk factors	49
Dependence on publishers and their advertising practices	49
Smart contract limitations	49
Regulatory risks	49
Price of Bitcoin and Ethereum	50

Rapid changes in technology may adversely affect mining business	50
Fluctuation in token benefits	50
Sales and other taxes	50
Force Majeure	51
Compliance with U.S. laws and regulations	51
Disclosure of information	51
Value of 1WO Token	52
Other risks	52
References	53
Disclaimer	54

Introduction

Online content consumption across many, if not most, web sites doesn't align well with advertising practices. There is a deepening conflict between a good online User Experience that readers want and survival necessity for publishers who want to serve as much advertising as possible to make their business model work.

Like many other traditional industries, the media business is struggling and going through a major transformation powered by new technologies, implemented via new user experiences and supported by proper business models to drive such change toward a real and sustainable solution and not a solution that goes on to create another set of problems (as has happened with “native” and “programmatic” advertising approaches).

Among many challenges, the media industry and overall the online publishing, advertising, and internet services sectors are facing the following issues:

- **Retention of readers** who increasingly are switching to social media as their primary source of news and other daily information of interest;
- **Fast growth of ad blockers**, especially among Millennials and other technology-advanced readers in general that are not hesitant to install such browser plugins on their desktop and now mobile computing platform;
- Extensive **fraud problems** of monetization within the AdTech industry, usually attributed to bot traffic (sometimes called “arbitrage traffic”) that many publishers buy knowingly or not; this problem is increasingly causing brands and advertisers to redirect their budgets away from sites running just classical display ad campaigns, which in turn causes major financial setback for those relying on this business model;
- **Questionable practices** such as *Pay-to-play* (or Guaranteed CPM approach) that are causing asymmetrical relationships with providers of technology and causing short-term wins but long term losses for publishers who entertain them;

On top of this comes the most fundamental problem of online publishing: an eternal conflict between, on one hand, the need to create and maintain good content encapsulated within a good user experience (UX), and on the other hand the need to make money via advertising (especially programmatic) which is very hard, almost impossible, to balance and sites usually do well either one or the other, but rarely both simultaneously.

People who use ad blockers are now dealing with soft or hard warnings that content will be available ONLY if they turn off their blockers, which creates a “no win” situation between publishers and their audiences.

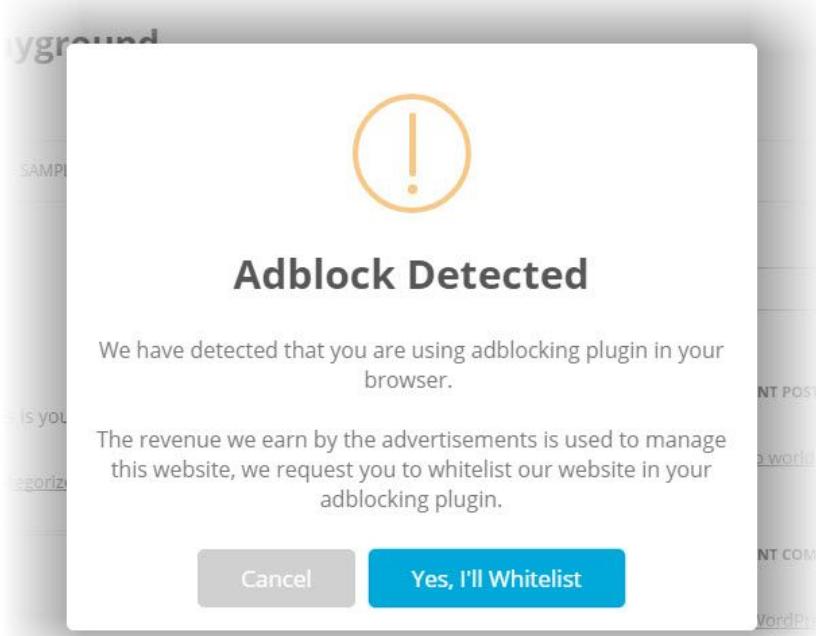
With the introduction of Blockchain into the Media industry, including the first wave of pilot implementations of various cryptocurrencies for attention, advertising, engagement etc., is certainly a step in the right direction and serves the ultimate goal of establishing a new type of relationship, where all three parts of the “Industry Triangle” (**Readers -- Publishers -- Advertisers**) are motivated and have proper incentives, enabling a true Win-Win-Win environment.

1World is a firm believer in motivating readers of the web sites to come visit more often via giving them incentives for their engagements and contributions, and offering a choice of being exposed or not to the ads on the site. All these elements have to come together to make the “Reader – Publisher -- Advertiser” triangle work.

The 1World platform has been providing the tools and the network for deployments since 2013, and the next step is further decentralizing it and introducing 1World (1WO) Tokens that will be circulating in this triangle and bringing benefits to all parties involved: namely, readers earn for engagement, publishers get new revenue streams, and advertisers receive discounts and access to wide-spread diversified and attractive inventory.

The future in this industry is decentralized models where the economy of interactions is built on very natural principles where people are legitimate active PARTICIPANTS with rights of their own rather than just being treated as cookie-tagged “subjects of advertising” in respective “digital prisons”¹ driven by a few monopolistic players. This future eliminates both monopolism of these behemoths and turns the digital economy back over to people

with proper experiences, ownership of their online identity, and incentives that bring more value to them.



The experience no reader really wants!

1World Vision & Execution

1World Online is a Silicon Valley company headquartered and operating in San Jose, California since 2012 with a mission to build a global software platform to supplement online content with interactive tools (such as Polls, Surveys, Quizzes, Insights pages, Interactive Maps and other formats) that help answer the question “*What Do People Really Think?*” This is done via contextual, relevant, organic, and highly engaging data collection with analytical insight on site and across network of participating sites.

We believe that any business objective in online services centered around content publishing should be accomplished via organic experiences where a conversation with readers has to be established first (e.g., via presenting a poll widget related to the article that the reader will be motivated to vote on and express their opinion), then the publisher learns more about this user (not only the immediate response, but also a history of previous responses to various questions and other engagements observed and captured by the 1World tools), so this analysis helps to provide better content (e.g., next question in the rotation or suggested articles to read next) and better promotion, such as a targeted advertisement or commercial mini-survey or some other type of online service offering.



This approach can be stated in three value-proposition words: “*Engage – Research – Promote*”, which is a progression model where all participating parties (Readers, Publishers and Advertisers) benefit simultaneously.

Here is an example of typical 1World Deployment (in this case: *San Francisco Examiner*) where the article is supplemented with a 1World widget automatically showing relevant poll questions to supplement the content and initiate a conversation between the reader and the

publisher plus an advertising component that also is contextual and better targeted for the specific reader based on context, history of engagements, and responses to poll questions.

The screenshot shows the San Francisco Examiner homepage from May 21, 2017. The main headline is "Bay to Breakers transportation and traffic survival guide". Below the headline is a large photograph of a crowded Bay to Breakers race. A sidebar on the right contains a poll titled "SF EXAMINER DECISION MAKER" about daily commute transport, with options for car, public transportation, or bicycle. At the bottom right is a thumbnail for a travel guide titled "5 Must-See Places In New York".

SAN FRANCISCO SINCE 1865

Sunday May 21, 2017

Bay to Breakers transportation and traffic survival guide



Thousands of runners make their way down Hayes Street during the annual Bay to Breakers race in San Francisco, Calif. Sunday, May 15, 2016. (Jessica Christian/S.F. Examiner)

By Joe Fitzgerald Rodriguez on May 18, 2017 4:15 pm

 Every year for the Bay to Breakers race, San Franciscans divide into two camps:

Those who grab a beer, don their tutu's and stroll the race, and those who avoid the race any way they possibly can.

To achieve either goal, a number of transportation agencies released guides for avoiding the

SF EXAMINER DECISION MAKER

What form of transport do you use in your daily commute?

LEARN MORE  Way out

A car
 Public transportaion
 A bicycle

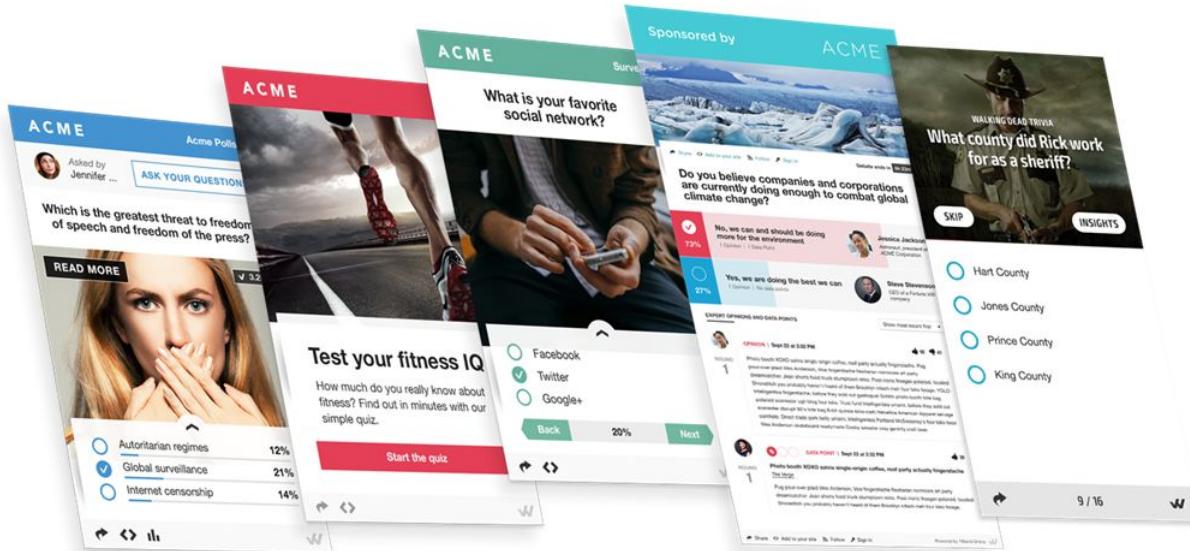
Powered by 1World Online

5 Must-See Places In New York

History of 1World Development and Deployments

Since its start in 2012, 1World has built and delivered a robust and comprehensive interactive platform that includes **front-end** tools, a **back-end** dashboard of analytics, and a **network** of participating publishers around the world. In addition, various organizations and brands are using 1World tools for interactive campaigns as well as ongoing audience engagements.

The front-end includes a variety of interactive, easy-to-customize tools such as Polls, Quizzes, Debates, Trivia Feeds, Surveys, and Interactive Maps. Each of them helps publishers and brands learn from and engage with their particular audiences to maximize engagement and increase revenue and efficiency of campaigns.



1World automatically produces **Smart Content** (such as Insights pages that include interactive maps and other analytics) that are linked to the modules (widgets) installed on partner sites. This content increases sharing over social networks, improves relevancy, and draws in new readers—with the core benefits of increasing audience and reach.



1World **Analytics** includes a suite of data analysis, research tools, and capabilities that provide insight into consumer opinions, preferences, and behaviors. 1World is also an IBM Watson Certified Partner and provides cognitive science powered analytics incorporating Watson.



1World contracted partners are Google, AOL, Sekindo, Matomy for Advertisement; CINT for Data (profiling and paid survey participation for users), IBM Watson for analytics and more targeted advertising and content recommendations.

1World Today

1World has well-developed products, business models, and deployments:

- Over **\$8M** USD equity capital raised from VC and Angel Investors in USA, Japan, India, Singapore, Europe, Hong Kong, Israel, Russia & Korea.
- Business model: Ads + SaaS / Licensing; **\$0.8M** in 2015, **\$1.3M** in 2016
- Over **12 Million** lifetime user engagements via the 1World platform
- Over **2500** registered partners of all types (publishers and brands)
- Over **46 Million** total votes collected via polls, surveys and other tools
- Up to **1.5 Million** monthly engagement widget participants
- 30 languages supported
- Up to **140 Million** monthly global impressions*
- **6** patents filed
- **5** industry awards
- **4** Offices Worldwide (United States, Ukraine, EU, Latin America)
- **Strategic Partners:** Google, AOL, IBM Watson, Amazon AWS, Cint, McCann

Market size and opportunity

According to DENTSU Aegis^[2] Global ad spend will reach \$563.4 billion in 2017 with digital driving growth at 3.8% amid cautious near-term outlook.

"We are reaching a tipping point in ad spend now as digital overtakes television, mobile overtakes desktop and paid search overtakes print. Digital and data must now be the default settings for advertisers. Evolving to people-based marketing rather than audience-based marketing and using data to increase addressability is essential for brands to manage tighter conditions in 2017 while positioning themselves for future growth."

-- Jerry Buhlmann, CEO of Dentsu Aegis Network

Combined with the overall size of online services this creates an enormous opportunity to address one of the biggest Internet behaviors, as a product, on an unprecedented scale.

	Year-on-year % growth at current prices		
	2016a	2017f	2018f
GLOBAL	4.8 (4.4)	3.8 (4.0)	4.3
NORTH AMERICA	5.0 (5.0)	3.6 (3.8)	4.0
USA	5.0 (5.0)	3.6 (3.8)	4.0
CANADA	3.1 (3.0)	3.1 (3.0)	2.7
W.EUROPE	4.0 (2.9)	3.5 (2.7)	3.6
UK	6.1 (5.4)	4.0 (4.6)	5.9
GERMANY	2.3 (2.3)	2.6 (2.1)	3.0
FRANCE	0.9 (0.9)	1.6 (1.2)	2.0
ITALY	3.5 (1.3)	0.8 (0.8)	1.5
SPAIN	6.8 (5.0)	5.0 (4.4)	3.6
C&EE	7.6 (4.7)	6.6 (5.5)	6.0
RUSSIA	11.4 (6.2)	9.8 (5.2)	7.8
ASIA-PACIFIC	4.7 (3.9)	4.3 (4.2)	4.6
AUSTRALIA	4.8 (5.4)	4.1 (4.5)	4.8
CHINA	7.4 (5.7)	6.0 (5.5)	5.4
INDIA	11.9 (12.0)	13.0 (13.9)	12.2
JAPAN	1.9 (1.8)	1.7 (1.2)	1.7
LATIN AMERICA	11.9 (10.0)	7.0 (9.8)	8.9
BRAZIL	5.4 (4.8)	2.1 (4.5)	5.0

Figures in brackets show our previous forecasts from September 2016

1World Intellectual Property (IP) Portfolio

1World has an IP patent portfolio it has been developing since 2012 in collaboration with top law firm Wilmer Hale, utilizing both their Silicon Valley and Boston offices. 1World files its patents first in the USA, and then usually pursues International filings, PCT applications to cover Europe and also files in select Asian countries such as Japan and China. The first five patents filed by 1World covered the basic 1World service and widget-analytics system and its applications for engagement, research, and content syndication and includes the following patent filings:

- **Chains of Polls as a unique data collection engine**
U.S. Patent Application No.: 61/841,022
- **Dynamic Analytics Research Engine / Normalization algorithm**
U.S. Patent Application No.: 61/841,118
- **Interactive Data Exchange Network**
U.S. Patent Application No.: 62/046,554
- **Crowdsourced Polls and their sharing in public space**
U.S. Patent Application No.: 2209340.00125US1
- **Insight Pages & In-place Analytics (3 claims)**
U.S. Patent Application No.: 158254438

In September 2016, three related patents were filed together (U.S. Patent Application No.: 2209340.00126US1US1) as a comprehensive provisional application, and we will pull out each one individually and file utility patents for each in September 2017. These three independent claims, with numerous dependent claims, are as follows:

1. Insights Page – a dynamic “second page” on web or mobile pages, along with new ad inventory

2. Smart Profiling & Data Presentation System -- creates a psychographic profile for each respondent on a 2x2 matrix, that then triggers intelligent presentation of data to each Web viewer.
3. In-place Analytics -- end-users and administrators can click to see analytics from within that particular widget, in situ.

This majority of 1World's patent portfolio is themed around inherent higher engagement for web publishers—or the measurement of this engagement-- based on relevant and supporting information to an initial and main content topic. The power of the 1World system is the interactivity and choice of navigation, choice of what to read or consume, and simultaneously the opportunity for web publishers of data to enhance the information gathering and understanding process, which in turn allows for ever-increasing engagement with the content. And in the case the Dynamic Analytics Research Engine-- a normalization algorithm patent (known as "DRP")-- the information gathering and understanding can provide scientifically sound research results as well; this research objective is further strengthened by the "Chains of Polls as a Unique Data Collection Engine" patent that focuses on the method of data collection over time and in different venues (web, mobile, print, etc.)

"Engagement" describes the interest in interacting with the digital content and interactive elements presented to end-users, such as voting on a poll in a 1World Poller widget or clicking on "Learn More" after one has voted. Content engagement is also associated with better comprehension and processing of the material, better enjoyment of the subject matter, and increased statistics of web viewing such as more time on the web site (section of the site, technically) and/or more monetization achieved by more, and more relevant, advertisements able to be displayed--and thus higher CPMs can be commanded.

1World's inventions listed thus help achieve higher engagement with the content being presented to the end-user. Looking just at the most recent patent filing in September 2016 to make the point, the additional Insight pages – and/or additional information in-situ within the spot on the web page (In-place Analytics) – of insights from the poll or other "interactive elements of choice" results, or putting web pages of data through a filter depending on their

psychographic outlooks and worldviews (Smart Profiling & Data Presentation System), and all three of these inventions can be applied simultaneously. The expected result of using these innovations is even higher engagement, a positive spiral where publisher editors, web site readers, and monetization goals all benefit.

Web publishers have a keen interest in raising engagement rates of their content. The shift in the publishing industry from print format to an online publishing medium has brought a revolution in the way news is produced and consumed. But along with this transformation of the publishing industry comes challenges. The low entry barrier, outdated tools, declining ad revenue, and the decimation of the once profitable classified sections of newspapers, and continuous change in search engine algorithms, combined with short-attention span of the online audiences has caused major industry problems. It has forced many online publishers to go out of business or desperately search ways to increase online engagement rates so as to stay in business, either by content subscriptions from avid fans (committed readers) of the content provider or online advertising to readers on their website or mobile app. These innovations listed above all are meant to address the problems facing web publishers by helping them increase their engagement rates with each individual end-user web visitor, and in turn generate more revenue and profits.

1World's Patent-pending DRP Algorithm

Normalizing Crowd-Sourced Votes



1World Cryptocurrency & Use Cases

1World Online will be introducing its own cryptocurrency called **1WO** (pronounced as *one-world*).

1WO Token is an internal cryptocurrency circulated inside the ecosystem that 1World Online has created and continues to grow. All payments inside such an ecosystem will be conducted only via 1WO Tokens. For a convenient exchange of 1WO Tokens into fiat and various popular cryptocurrencies there will be references to external marketplace such as Coin Exchanges where 1WO coins will be listed after the ICO.

The issuance of 1WO currency is capped at 160 Million tokens and no additional issuances are planned.

From the very beginning the issued token mass is designed to support the growing volume of transactions inside of the ecosystem during the scaling, which implies ongoing organic growth of the 1WO Token buying power.

The tokens could be split up to 8th decimal point position, so the growth of their value won't be a problem for processing transactions of any viable size within 1World Online ecosystem.

The following base use cases covering all three participants of the *Triangle*: Readers ➔ Publishers ➔ Advertisers

Readers' Use Cases

End users on the site can earn these tokens by collecting points for engagements and contributions. First these earnings are coming in the format of points, which are already implemented in the 1World Platform and have functioned for years as a standard feature reflecting the amount of user activities. Upon reaching a designated threshold, these points will be converted into 1WO Tokens as soon as the user registers and creates a wallet.

Then spending and use of earned tokens can be done either inside of the 1World system or outside of it. Inside, users will be offered a few pre-defined choices:

- a) Use 1WO coins for **micropayments to disable advertising** and still enjoy the content. This case solves the fundamental problem when readers with Ad Blockers are warned by the site they are visiting that content isn't available until ad blocker is turned off or a subscription payment received. 1World will make such micropayments seamless and establish a proper user experience that works for both the publisher and the reader.
- b) Use 1WO coins to **purchase Advertising space** and / or **initiate Data collection service** (Mini-surveys) via 1World widgets. This operation is handled via a built-in Marketplace with one of our partners, CINT.
- c) Redeem 1WO coins at the **online store** that accepts tokens for their goods via an easy one-click transition from their 1World User Profile to the respective Landing Page;
- d) Take earned 1WO coins outside of the system, e.g. to a **Coin Exchange** that accepts 1WO tokens and sell them there (Secondary Markets).

1World's current detailed scoring (points-based) system is described in the next section.

Conversion from points to tokens will be different for different sites and determined by a **COT (Coefficient of Transformation)**. It will depend on site reputation, volume of traffic, and related factors. Initially COT will be established by the 1World Team with a help of Advisory Board that includes top media experts. Then an algorithm will be developed, as indicated in the 1Word Roadmap, to make such calculations and adjustments automated and well-balanced.

Publisher's Use Cases

Publishers will be accepting 1World Tokens (coins) based on the following motivation and principles:

- Sell remnant inventory for tokens (as otherwise there is zero dollars from it)
- Enable readers to earn and spend tokens, thus drastically increasing retention rates
- Provide an option to allow 10% of Ads revenue received via 1World service go to readers, and 1World will match it (subject of approval)

On top of all this, publishers will be receiving all of the standard benefits associated with the use of 1World's Platform such as increase of User Engagement and growth of key metrics such as time on site and number of visited pages, Smart Content (such as Poll results with Interactive Maps, Infographics and other auto-generated Insights), and advertising revenue.

Advertiser's Use Cases

Advertisers will be paying for running ad campaigns with 1World Tokens (coins) or fiat (such as US dollars) based on the following principles:

- Get discounts for using 1WO Tokens (versus cash or other equivalents)
- Have access to 1World network of sites (clusters) supporting 1W widgets
- Have the ability to ask questions, receive feedback during the campaign
- Have ability to profile their audience thus increasing targeting efficiency

Mechanics of work will be as following:

1. At the entrance to 1World system, there is a marketplace where a client (brand, agency or an individual) can buy tokens for Fiat and Crypto currency. These tokens could be also earned via engagements (also called “mining”)
2. Prices for advertising in the system are set in dollars and are driven by publishers, but aggregated and presented by 1World
3. To pay for accommodation, client must purchase tokens and pay for them in accordance with the current exchange rate of the token to the dollar
4. Once tokens are purchased, campaign is launched per purchased volume of impressions and targeting, all driven by 1World platform and its campaign management system;
5. Upon campaign completion, an automatic report is generated and delivered to the client;
6. Tokens for the campaign now are in possession of the Publisher and 1World based on the business rules; usually it's Revenue Share model, possibly reflected in Smart Contracts;
7. Publisher has a choice to hold Tokens upon campaign completion (if they expect their value / price to grow) or use an external exchange (conveniently plugged in) to convert to fiat / other crypto currency.

** During the campaign users who get involved receive respective rewards with Tokens at the current token exchange rate to the dollar and are calculated based on dollar prices*

The volume of emission of tokens is strictly limited to the framework set forth in the ICO.

With the growth of the system's turnover, the price of the token is expected to grow naturally; the original limited mass of tokens will be providing the foundation for an increasing value of transactions.

Thanks to the growing token exchange rate, we expect to address the challenge of queues for advertising placements. The advertiser, who has tokens, can pay for their accommodation right away or wait for these tokens to grow in price in order to get more volume of placement.

In addition, the total capitalization of the token-mass (in dollars) is expected to grow, which in some ways makes this ICO a vehicle for bringing continuous benefits to token-holders.

Why Blockchain?

There are many reasons why 1World platform transition to use Blockchain is the right strategic path and a very organic and natural fit for the product developed and deployed since 2012:

1. 1World is already implemented as a **distributed** and partially **decentralized** system where 1W widgets are running on many sites, in many languages and many countries and support sophisticated mechanisms if information exchange, e.g. ability to do “Compare & Contrast” of public opinion on polls that are tagged as “public” and are available to many audiences on many sites. Now the platform is transitioning to even more powerful model where established mechanisms can operate in peer-to-peer mode and have less (or eventually no) dependency on central server.
2. 1World already has a **scoring system** used primarily for gamification, that is based on engagement and contributions by site audiences. Transitioning this system to Tokens makes incentives more material and leads to wider acceptance of all ecosystem participants.
3. There is a huge benefit to use **Blockchain ID** to identify users in the 1World ecosystem instead of relying on cookies, an old and increasingly unreliable technology, especially

for readers who start their engagement anonymously and they decide to register in order to earn tokens and, at some point, to redeem them.

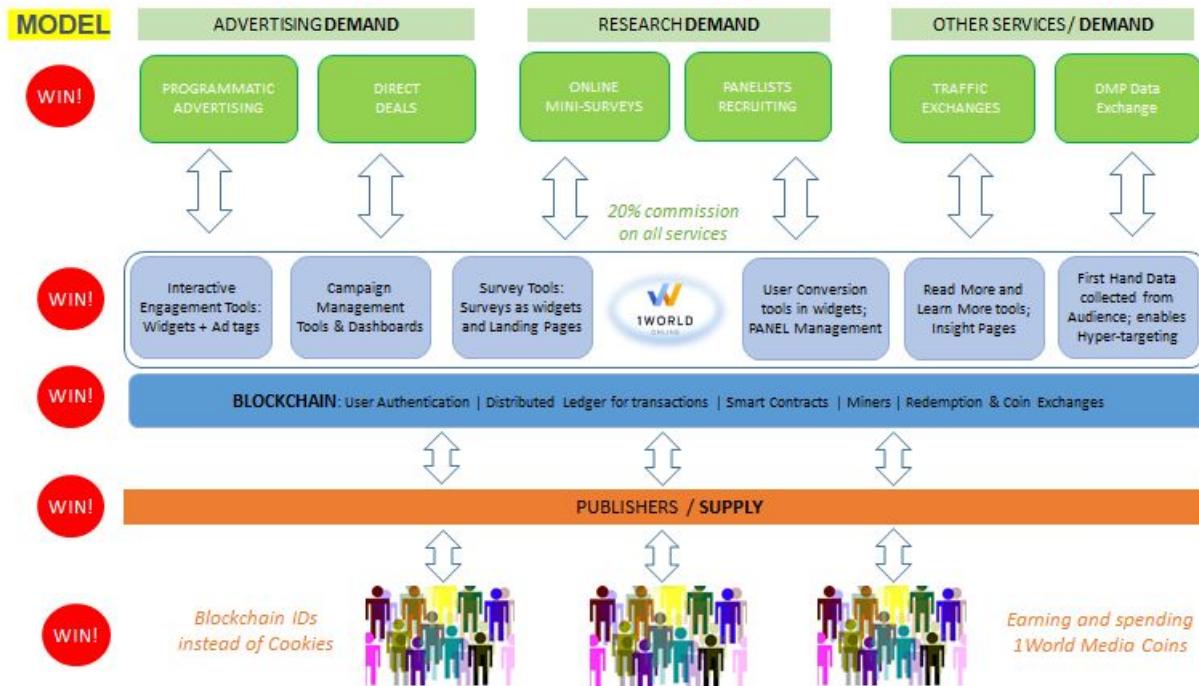
4. 1World connects services, such as advertising and research to audiences and is essentially serving as a layer to facilitate many ***marketplaces*** and many competing offerings coming from these marketplaces. Using Smart Contracts for such operations makes the platform operation robust and transparent and therefore putting it on the public decentralized ledger is the best approach to grow the business.

Overall there is fast growing set of Blockchain-based services, applications, and technologies so it make sense to have it utilized not only for the immediate opportunities, but also for future use cases.

The 1World Economy

The 1World decentralized economy is already developed and will be built further around the concept of connecting audiences on variety of websites with services from various providers, including, but not limited to:

- **Advertising:** (programmatic ads and direct deals)
- **Research:** via Commercial Polls and “Mini-Surveys”
- **Traffic:** tools for generating traffic to and from other sites
- **Data exchange:** passing information to third party analytics systems, such as DMP
- **TBD** future services that also can be routed via 1World widgets



1World Points System Today

1World's Loyalty and Gamification system used since 2013:

N	Action Description	Current Points	Comments
1	Login to the system	5	
2	Initial Registration	100	
3	Profile Management	10	
4	Vote on Poll	10	
5	Vote on Opinion	10	
6	Vote on Data Point	5	
7	Contribute Poll	15	
8	Contribute Data Point	25	
9	Contribute Opinion	50	
10	Opinion Voted Up by Member	10	
11	Opinion Voted Up by Expert	20	
12	Opinion Voted Down by Member	5	
13	Opinion Voted Down by Expert	5	
14	Sharing to social media by user	20	
15	Click on Read More or Learn More button	15	

1World Wallet

The 1World Wallet will be established for all registered users and allows them to earn and spend their tokens within the 1World ecosystem

Expansion with More Use Cases

It is expected that 1World will be adding new types of engagements and new associated services that will run inside deployed widgets and give even more opportunities to earn and spend points and then convert the monetary gains to 1WO Tokens. This type of new operations will be supported in future releases of 1World software.

1World Token Issuance

1World Online will be issuing 1WO tokens according to the following plan:

Token Sale Volume	<ul style="list-style-type: none">• 50M worth of 1WO Tokens cap for this ICO (also time limited to the ICO completion)• Total plan is to issue up to 160M worth of tokens in 3 years
Token Issue Volume	Up to 160 Million tokens lifetime cap
Minimum Target	Minimum ICO target will be set as \$5M
Distribution of Tokens	Covered in the next section
Token Price at Issue	\$1.00 USD at a start, with \$1.20 as upper limit towards the end of ICO
Website link	https://ico.1worldonline.com
Accepted forms of payment	BTC, LTC, DASH, ETC, ETH, XMR, ZEC or USD (via wire transfer)
Presale Start Date	September 7, 2017, 12:00 PM PDT <ul style="list-style-type: none">• <i>Discount 25%</i> September 7 - September 14• <i>Discount 20%</i> September 14 - September 21• <i>Discount 15%</i> September 21 - September 28• <i>Discount 10%</i> September 28 - October 5
Public Presale End Date	October 05, 2017, 12:00 PM PDT
ICO Start Date	October 05, 2017, 12:00 PM PDT
ICO End Date	November 06, 2017, 12:00 PM PDT
Token Issue Date	November 16, 2017, 12:00 PM PDT

1World Token Allocations

During the ICO for each one (1) Token sold in open sale:

- 1 Token will be issued for mining to Publishers and Partners
- 0.2 Token will be issued to the Team and Advisors / Bounty program
- 1 Token will be allocated and placed to Reserve (aka, Liquidity pool)

As a result, allocation of 1World 1WO Tokens will be as following:



Escrow

Minimum ICO target will be set as **\$5 Million**

- If less money is raised, then all of proceeds will be returned within one month after the ICO completion

- If a force majeure occurs in tokens issuance, funds will be returned to buyers within two months

For the 1World ICO program, an escrow account will be set-up and utilized, managed by a Los Angeles-based provider Blockchain Law Group

The funds will be released from escrow to 1World Online upon hitting the \$5M target and after the distribution of tokens.

Supply Schedule

Initial Token supply will be a maximum of up to 110 Million tokens (50 Million direct sale + 50 Million for mining and 10 Million for the Team and Advisors) and will be distributed upon ICO completion on or around November 16, 2017.

Periodic Token transition out of reserve to support the company growth starting from Year 2018 will be based on the business needs and tokens will be either sold on open market or allocated to publishers and partners for their audiences to mine them via engagement. It will be done per the same rules as outlined earlier for the original ICO in Year 2017 via the Scoring system and COT (*Coefficient of Transformation*) adjusting conversion of earned points to 1WO Tokens per publisher.

In order to sustain a robust economy for its tokens and to facilitate token value growth 1World plans to launch a new Tier 1 media holding company (a publisher with many digital sites and multi-million audiences) every quarter to support this economy. This approach will support the “smooth token growth” model with a virtual “corridor” established.

Payments Processing

Token buyers must register at tokensale.icobox.io, entering and confirming their email address. After the registration, users gain access to their personal accounts at tokensale.icobox.io where they will have separate wallets for BTC, ETH, LTC, Dash, Zcash, ETC, or USD. In their accounts users can choose the desired number of 1WO Tokens and

transfer the required payment amount in one of the accepted cryptocurrencies or generate an invoice for a wire transfer.

Once the payment is received, funds will appear in the corresponding wallet in the user's account and may be used for purchasing 1WO Tokens. Until the token purchase is made, the funds may be withdrawn from the account at tokensale.icobox.io at any moment by sending a request to support@1worldonline.com. Tokens are purchased at the price in effect at the time of purchasing, not at the price in effect at the time when funds have been sent or received by the platform. When buying tokens with currency other than BTC, the exchange rate is fixed at the time of token purchase.

Accounts at tokensale.1worldonline.com will be accessible several days before the start of the 1WO token sale. Users will be able to sign up and make transfers to their 1WO accounts, but will not have the option to buy 1WO Tokens with deposited funds until the start of the 1WO Token presale. To take part in the public presale, a buyer will need to purchase at least the specified minimum number of 1WO tokens (10,000 or 1,000 1WO Tokens, depending on the date and time of purchase). The general 1WO Token sale has no minimum entrance threshold, except for the minimum transfer amount specified by the relevant blockchain or bank used by the buyer.

After the token purchase is complete, the information about 1WO Tokens credited to the 1WO buyers should appear in their accounts at tokensale.1worldonline.com immediately.

Token Distribution Event

After the 1WO Token sale is over, a personal account will be automatically generated at ico.1worldonline.com for every 1WO Token holder. 1WO tokens will be issued and transferred to these accounts.

Once this step is complete, 1WO Token holders may at any time transfer their 1WO Tokens to any third-party ETH wallet supporting ERC-20 standard.

BTC and wire transfer proceeds from the 1World 1WO Token sale during the ICO Sale will be deposited in escrow where they will be kept in BTC and USD. Supported original payments made in other cryptocurrencies will be accumulated, converted to BTC, and also deposited in escrow on a regular basis.

Token Distribution for the Team

After the 1WO tokens become available to the team (7% of overall allocation, which is 20% of the original open sale), they will be distributed as following:

- 1World Core Team: up to 50%
- Advisors & Key ICO Contributors: up to 20%
- Bounty Program: up to 10%
- Reserve: Up to 20%

** 1World reserves the right to direct these funds to certain areas based on specific ICO development to maximize the efficiency and increase the Token sales.*

Use of the Proceeds

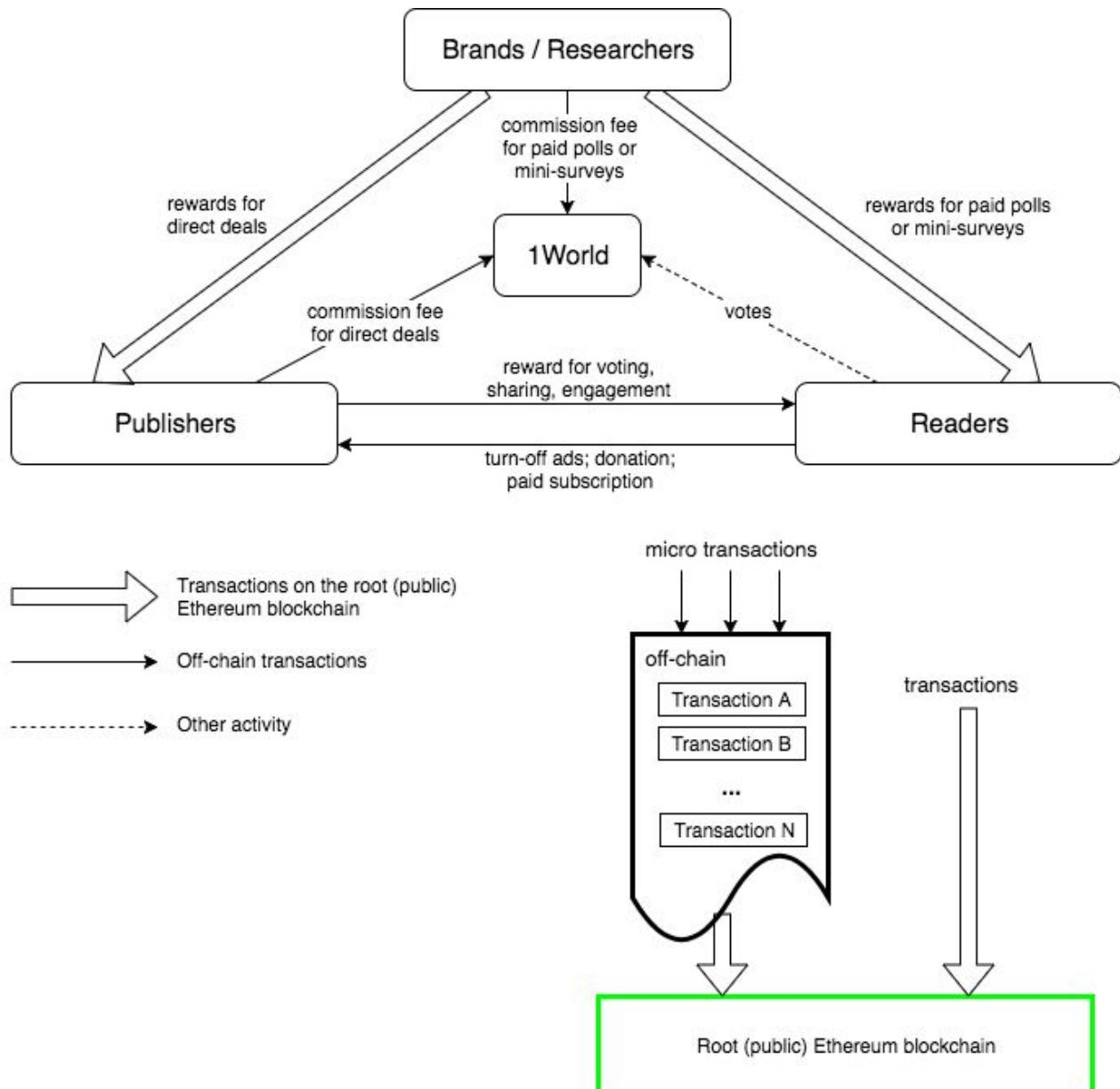
1World plans the following use of the proceeds collected in the ICO Open Sale:

- Up to 30% - Core US Team: Business / Marketing / Customer Success
- Up to 25% - R&D and Technical support
- Up to 20% - Global Expansion
- Up to 10% - Legal, IP and other services
- Up to 5% - PR / Events / Promotions
- Up to 10% - Reserve / Miscellaneous

** 1World reserves the right to direct these funds to certain areas based on business opportunities and market situation with the goal of growing both business and company valuations in mind,*

Technical implementation

Technical implementation aspects of the 1World economy and cryptocurrency are outlined in the following diagram:



At the beginning 1WO Tokens will be implemented as a combination of public on-chain Ethereum blockchain, and off-chain solution like Raiden, Plasma or State channels (or private blockchain solution like Exonum).

Users (brands, publishers, researchers, readers) who want to transfer 1WO Tokens will be able to interact with the public Ethereum blockchain, which will serve as the currency's decentralized settlement layer.

Implementation Based on Ethereum ERC20

The Ethereum blockchain is currently the industry standard for issuing custom digital assets and smart contracts. One of the main reasons is using existing Ethereum's infrastructure instead of building an entirely new blockchain. The latest and most popular ERC20 token interface is the best choice compatible with the existing infrastructure of the Ethereum ecosystem, development tools, wallets, and exchanges. Also ERC20 allows for seamless interaction with other smart contracts and decentralized applications on the Ethereum blockchain.

Current Ethereum Limitations

Blockchain in general (and Ethereum blockchain in particular) is a decentralized network. Decentralization is critical because it eliminates single points of failure or control that makes downtime, censorship, fraud or third-party interference impossible. These advantages and active ecosystem make Ethereum a good choice for the 1World decentralized economy. But, on the other hand, the nature of blockchain decentralization (Signature verification, Consensus mechanisms, Redundancy, etc.) leads to some disadvantages in comparison with centralized databases. The biggest Ethereum disadvantage for proposed 1World economy is Ethereum scalability.

The well-known fact is the current Ethereum Proof-of-Work blockchain has scalability bottlenecks:

- a fee to be paid for every single transaction
- a transactions confirmation takes tens of seconds in average
- a number of transactions per second (transaction throughput) that is small enough to process increasing number of applications

These limitations make micro-transactions, such as the ones the 1World Online platform performs for typical use cases, not economically viable to put on the public ledger and be charged. An example would be 1WO Tokens to reward end-users on the site for their engagements and content contributions.

If a public Ethereum blockchain is used ‘as is’ for these transactions, then:

- a fee for transaction would be comparable or even higher than reward amount itself;
- end-users would wait for tens of seconds until transaction is confirmed;
- huge number of 1WO transaction would congest the Ethereum’s throughput capability.

That’s why 1World Online could not use the existing version of public Ethereum blockchain ‘as is’ for some kinds of transactions and has to find some solution. There is no silver bullet here and now, but Ethereum is a new technology, a lot of efforts are made to solve these Ethereum’s limitations. For example, the core Ethereum team understands the limitations and is trying to solve them: a recent Ethereum Metropolis hard fork is a first step in transition from Proof-of-Work to Proof-of-Stake. Moreover a lot of other solutions on the top of Ethereum are in progress. And most experts agree that this is only the question of time: some solution or combination of solutions will be developed as is often the case with evolving technology.

This concerns not Ethereum blockchain only. The similar issues for the Bitcoin blockchain, a crypto currency network older than Ethereum, have been successfully solved by [Lightning Network](#).

There are several solutions that are going to solve these blockchain bottlenecks like [Machinomy](#), [Toy State Channels](#) and other, but a few of them are worth to be mentioning in detail:

- Raiden network
- Plasma
- Smart contracts
- Exonum

Raiden Network

[Raiden network](#) is similar to Bitcoin's Lightning Network - off-chain high speed asset transfers network for Ethereum ERC20 tokens that has been under development now for nearly two years and is close to release. It uses a network of off-chain state channels to securely transfer any ERC20 token without the requirement to transact on the blockchain. The Raiden Network will provide a payment system based on payment channel technology that scales with the number of its users: the bigger the Raiden Network becomes, the higher its maximum throughput will be, with practically no upper limit in sight.

One more important thing: unlike Ethereum public transactions, Raiden transfers will be private between the payer, the payee, and the nodes forwarding the transfer. When channels are settled, only the sum of transactions will become visible to the entire world.

Plasma

The [Plasma](#) solution was recently announced by Joseph Poon and Vitalik Buterin; it is a framework that is scalable to a significant amount of state updates by using the Proof-of-Stake method. Plasma works on top of a root (main) Ethereum blockchain. Its idea is to eliminate the need to generate on-chain transactions for every state update by leveraging coalesced state updates that are comprised of the bitmaps in which the transactions are composed. The root network contract processes only a very small number of commitments

from child blockchains that are able to do a large number of computations. Commitments are broadcasted periodically to the root blockchain from the child blockchains.

State Channels

In addition to open-source solutions like Plasma or Raiden, 1World Online could implement their own solution that utilizes State channels for appropriate transactions.

State Channels are a design pattern in which signed messages representing transactions are exchanged between two parties. Any party can choose at any time to broadcast the last message to the blockchain, updating it to a state representative of the sum of all exchanges between the two parties.

State channels are an important technique for allowing some blockchain operations to be conducted off of the blockchain, while retaining or even improving the underlying security guarantees that a blockchain offers.

In particular, state channels can be used to significantly improve the performance of interactions between a designated group of people, while significantly reducing transaction costs. They can be applied to payments, smart contracts, and many other scenarios. [9]

Exonum

[Exonum](#) is an open source enterprise-grade blockchain framework developed by the Bitfury Group. It allows to build a private or permissioned blockchain. At the moment, Exonum-based smart contracts can handle 3,000 transactions per second with a clearing time of 2.5 seconds (and up to 15,000 transactions per second in custom situations), that is significantly faster than any other public Blockchain networks. It uses custom-built Byzantine Fault Tolerant consensus algorithm that excludes any single point of failure, making blockchain resilient to nodes crashes or bad actors, without needing to mine blocks.

Exonum seems to be a perfect solution for 1World's micro-transactions, but there is a one significant obstacle: it anchors (periodically saves a cryptographic hash reflecting the state of

Exonum Blockchain) data to the Bitcoin blockchain in order to protect data against history revisions. But 1WO is Ethereum token and thus this will require to develop some extra module that allows to use Exonum together with Ethereum blockchain.

Smart Contracts

1World will utilize Smart Contract for appropriate transactions, both for Engagement and Services Delivery (Advertising and Research) use cases as described in this document.

Current Implementation Plan

It is highly expected that there will be others solutions that also could be used for 1World Online purposes. The common idea is to switch from saving all transactions on the public blockchain (on-chain) to saving them off-chain but ultimately enforceable on-chain. Moving transactions off of the chain leads to significant improvements in cost and speed. Anyway, there is a strong confidence that there will be at least one solution on top of Ethereum in the near future that allows highly scalable, low latency, and cost effective decentralized systems.

Use Cases for 1WO Tokens

For further consideration, it makes sense to separate 1World transactions based on their size, quantity, and the role of involved persons into:

- Reader micro-transactions to earn 1W points / 1WO Tokens
- Reader micro-transactions to spend 1WO Tokens
- Brands or Research related transactions
- Publisher's transactions

Reader (end-users) micro-transactions to earn 1W points / 1WO Tokens

Micro-transactions in 1World widgets on publishers' websites happen when site visitors (readers) earn 1WO points for:

- voting, sharing, click Read-More or Learn-More, or other related engagement actions;
- poll creation or other contributions;
- voting commercial polls or taking commercial mini-surveys;
- watching video ads;
- sign up to 1World Online by using email address or social sign-in (Facebook, Twitter, etc.);
- entering any of their profile info (gender, age, marital status, etc);

1World widgets will display how many 1WO points have been collected by current users, explain their value and suggest converting them to 1W Tokens. In order to convert points to tokens the user needs to:

- earn 500 (subject to refinement) 1WO points;
- sign up to 1World Online by using email address (and verify email address) or social sign-in (Facebook, Twitter, etc).
- Enter his/her wallet address

Conversion from 1WO points to 1WO Tokens will be made by conversion rate.

This rate is not fixed and depends on:

- 1WO Token price at the moment of conversion;
- a special publisher coefficient that is calculated by 1World Online algorithm based on its content, audience, etc.;
- a special geo-coefficient for the user's location;
- a time period when the points were earned.

After the conversion the user gets access to his/her own ‘My Wallet’ page with current balance, transaction history, etc. Also all future 1WO points earned, once they reach the established threshold, are immediately converted to 1WO Tokens and added to user’s wallet.

Because of current Ethereum PoW limitations (described above) all these micro-transactions are saved off-chain but ultimately enforceable on-chain by using Plasma, Raiden, Exonum or similar solution.

Reader micro-transactions to spend 1WO Tokens

Micro-transactions also happen when readers spend 1WO Tokens points for:

- convert 1WO Tokens to perks;
- pay publisher so as to not see advertisements during a week or a month;
- pay publisher to get access to paid content;
- donate to the author of the blog or some particular article.

All these micro-transactions are also saved off-chain but ultimately enforceable on-chain.

Brands or Research related transactions

Another type of transactions happen when brands or researchers spend 1WO Tokens for:

- running direct deals campaign
- running commercial polls or commercial mini-surveys

These transactions will be made from 1World Online Client Dashboard as a public Ethereum on-chain transactions because:

- the transaction amount is big enough when compared with the transaction fee
- brands or researchers could wait tens of seconds for a transaction confirmation
- they will happen much less often than readers micro-transactions and will not affect Ethereum transaction throughput significantly

Publishers transactions

Publishers earn 1WO Tokens for:

- their inventory used for direct deals campaigns
- their inventory used for commercial polls or commercial mini-surveys
- not displaying advertisement to some particular readers
- providing access to paid content to some particular readers

In case of direct deal campaigns transactions will be made in a public Ethereum on-chain.

All other transactions are saved off-chain but ultimately enforceable on-chain by using Plasma, Raiden, Exonum or similar solution. In addition to standard enforcing methods provided by off-chain solution, the off-chain transactions will be enforced to on-chain when there is a big enough 1WO Tokens for this particular publisher.

1World Reward Engine

The 1World Reward Engine will follow the implementation already done and released in version 2.x.

1World Campaigns Management

The 1World Campaigns Management implementation will be part of Release 3.0.

Identity Management & DID

Identity Management & DID will be further researched and released in future versions of software.

Summary

At the beginning 1WO Tokens will be implemented as a combination of public on-chain Ethereum blockchain, and off-chain solutions like Raiden, Plasma or Exonum.

Roadmap

1World continues its agile software development process and produces bi-weekly production releases with new functionalities as well as scheduled quarterly major updates with significant new product features.

The following roadmap items are planned for Years 2017-2020*

Q3 2017	<u>Core:</u> Blockchain integration: infrastructure <u>Apps:</u> Scoring visibility in 1World widgets
Q4 2017	<u>Core:</u> 1World Points system linked to 1WO Tokens <u>Apps:</u> User profile page to integrate Wallet & Redemption
Q1 2018	<u>Core:</u> Campaigns management for advertising and other services <u>Apps:</u> In-place analytics in widgets for users and admins <u>Network:</u> Vertical-focused clusters of publishers
Q2 2018	<u>Core:</u> Interactive Content Creation marketplace <u>Apps:</u> 1World Feed to integrate data sources & scoring
Q3 2018	<u>Core:</u> Interactive Media Unit (IMU) implementation <u>Apps:</u> 1World Button app

Q4 2018	<u>Core</u> : Research tools: panels management <u>Apps</u> : Mini-Surveys in 1World widgets <u>Network</u> : Integrating into online research marketplace
1H 2019	Blockchain 2.0 Integration / high scalability solution
2H 2019	A.I. driven Content Generation and linking to articles <u>Network</u> : Integrating into traffic generation marketplace
1H 2020	Global Syndication Network
2H 2020	A.I. driven Recommendation Engine

* Roadmap items are subject to change based on business opportunities and market trends

1World Technology Advisors

1World Blockchain Advisory Team



Vasiliy Suvorov
CTO Luxoft,
Technical Head
CryptoValley
Blockchain Group
Switzerland, EU



Alison Davis
Chairman of
Advisory Board,
Blockchain.capital
Board Director, Royal
Bank of Scotland
Silicon Valley, USA &
London, UK



Takatoshi Nakamura
Japan Society for
Security, Blockchain
2.0 & Security Expert
& Inventor
Tokyo, Japan



Ayako Miyaguchi
Founder of Japan
Blockchain
Association
USA & Japan



Alex Yastremsky
General Counsel,
Bitfury
ICO Expert
San Francisco, USA

US Business Advisors



Phil Yin
Managing Partner
Newsroom
Investments
Seattle, WA



Dr. Augie Grant
Professor at
University of South
Carolina
Ex -1World, 2Wire
Popular Author &
Researcher



Dmitry Kustov
CPA, owner of
BitTax.io
Principal at K&A
Expert in ICO
Taxation



Sean Koh
Owner Koherent
Records
US – South Korean
Entrepreneur &
projects coordinator



Peter Saulinier
Managing Partner
GCH Partners
New York, NYC



Ali Raheem
Partner / Principal
Ernst & Young
Silicon Valley, USA



Marios Anapliotis
**COO Open Bouquet,
1World Tech Advisor**
Silicon Valley, USA



Tina Ghataore
Band of Angels
EKKAM Marketing
Silicon Valley, USA



Lilia Shirman
**Sr. Advisor &
Marketing**
Head of Golden
Seed Group
Silicon Valley, USA



David Drake
Family office LDJ Capita
Crypto advocate. GP &
LP investor in real estate
and hedge funds.

Silicon Valley



Matthew Le Merle
FIFTH ERA, Keiretsu
Capital Keiretsu Forum



Ted Sanford
FlashSoft (acquired by
SanDisk), Baccel,
Infravio, AppStream



Michael Minkevich
Luxoft (LXFT) – IPO in
2013, StarForce
Technologies, Novosoft



Kira Makagon
RingCentral (IPO in 2013), Red
Ariil (acq. by iCrossing), Scopus
(IPO), Octane Software (acq. By
Epiphany)

New York Media Companies



Greg Kahn
Meredith Xcelerated,
Publicis, Omnicom



Dan Reich
Forbes, Harvard Business
Review, BuddyMedia
(acq. by SalesForce)



Greg Osberg
CEO Revlyst
Ex-CEO Newsweek
Philadelphia Media
Network, CNET



Jon Bond
Co-chairman of The
Shipyard Inc. and Chief
Tomorrowist at
TOMORRO.

Global Partners



David Schlesinger

Tripod Advisors, CCTV News, Thomson Reuters (Chief Global), Global Editors Network



George Hara

CEO of DEFTA Partners
Special Advisor to the Cabinet Office of the Prime Minister of Japan



Diana David

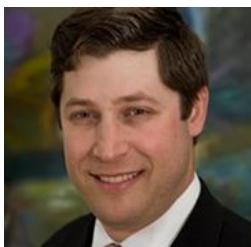
Vice President Financial Times, Advisor to Nest.VC in Hong Kong



Namho Chung

President of Keiretsu Forum Seoul, Korea Chapter X-Venture Labs

Industry Partners



Nathan McDonald

CEO Keiretsu Capital,
Head of Keiretsu North-West



Piyush Puri

VP of Brand Capital Worldwide,
Times Group of India



John Ricci

Head of US Angels,
MG Capital, Crimson Growth



James Zhao

CEO, Concept Art House and Spellgun,
Expert in Gaming Industry

1World Team

Over five years of the company's life we have built a globally distributed, very skillful, and dedicated team that combines technical, business and customer management / support skills and has demonstrated success in many deployments and projects.

1World Leadership Team



Alex Fedosseev

**President since 2011
CEO & Founder**

Ex 2Wire, 4Home,
3Com, Motorola
Mobility / Google,
Broadband Forum.

Brad Kayton

**Board Member
since 2011
Secretary; GM Data
and IP**

Ex 2Wire, 4Home,
Polycom, Zoom,
Vgo, Prizmiq, Serial
Entrepreneur

Kyoko Watanabe

**Board Member
since 2015**

Director of DEFTA
Partners;
Director of World
Alliance Forum;
Board Member of 5
portfolio companies

Vladimir Tyurenkov

**Board Member
since 2015**

Managing Partner
Steltec Capital;
LP at Hive Big Data,
Ex- Hansberger
Global Investors.

Neville Tarapowalla

**Advisory Member
since 2016**

Managing Partner at
Brand Capital/ BCCL,
Head of Times
Group USA; ex
Yahoo, Microsoft,
ex-CEO of Publicitas
Digital

1World US Business Team



Matt Ganeles

**1World West Coast
Project Manager
Blockchain
Ecosystem
Palo Alto, CA**

Katia Kourtseva

**1World West Coast
Finance & Office
Manager Logistics /
General support
Palo Alto, CA**

Emily Kayton

**1World East Coast
Content & CS
Manager Support
Publishers &
Brands
Boston, MA area**

Klaudia Kostarelas

**1World West Coast
Customer Success
Manager
Publishers, Brands,
Partners
Palo Alto, CA**

Quinn Miller

**Marketing Graphic
Designer
Palo Alto, CA**

1World Global Business Team



Nikhil Shah
India Representative
Times Group AM
Ahmedabad, India



Namho Chung
Korea Representative
Bus Development
Silicon Valley, USA
Seoul, South Korea



Daniella Franchin
Latin America Representative
BusDev and AM
San Paulo, Brazil



Mike Tanji
DEFTA Partners 1World Japan Representative
Yokohama, Japan



Sergei Makedonsky
Forrester Research
EEU in4media
Russia Representative
Moscow, Russia

1World Engineering Core Team



Valentina Volotskaya
R&D Executive Director
Lviv, Ukraine



Dmitry Volotskoy
Architect and ENG Lead
Lviv, Ukraine



Lera Kulikova
UX/UI designer
St. Petersburg, RF



Dmitry Birukov
Back-end Architect
St. Petersburg, RF



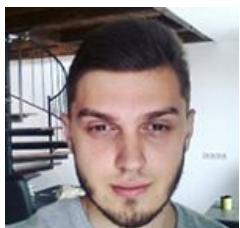
Nazar Pilipyak
Front-end Engineer
Lviv, Ukraine



Dmitry Mironov
Lead QA Engineer
Lviv, Ukraine



Eugene Slesarchuk
DevOps Engineer
Stary Oskol, RF



Robert Kovtiuk
Back-end Engineer
Lviv, Ukraine



Natalia Gordienko
Ad Operations
Kiev, Ukraine

1World Support Team



Ruth Lee

**Marketing &
User Experience**

Palo Alto, CA

Jongseung Lim

**Community
Manager**

US & South Korea

Austin King

**Community
Manager**

US & EN countries

Leonard Jackson

**Support &
Promotion Manager**

Rowland, NC

Jameel Bsata

**Support &
Promotion Manager**

Edmonton, Alberta,
Canada

1World Publishers and Partners



REUTERS

THE
HUFFINGTON
POST



THE TIMES
GROUP India

Forbes



Singapore Press Holdings



EL UNIVERSAL

oneindia
India's #1 Language Portal

THE ECONOMIC TIMES



EL UNIVERSAL
CARACAS, viernes 23 de abril, 2014 | Actualizado hace 5



NEWSOK

GANNETT
DIGITAL

LA
TRIBUNE

and many more..



Microsoft



greenWAVE
systems



WAFSF



BE STRONG
STOP BULLYING



MIRANTIS

Publishers using 1World

Examples can be found here: <https://ico.1worldonline.com/examples>

Advertisers using 1World

Examples can be found here: <https://ico.1worldonline.com/examples>

Partners integrating 1World

Examples can be found here: <https://ico.1worldonline.com/examples>

Risk factors

The acquisition of Tokens involves a high degree of risk, including but not limited to the risks described below.

Before acquiring tokens, it is recommended that each participant carefully weighs all the information and risks detailed in this White Paper, and, specifically, the following risk factors.

Dependence on publishers and their advertising practices

1World's business is dependent on existing advertising markets with its very complex infrastructure and existing policies and practices related to programmatic and direct advertisement, certain requirements for performance, reporting, discrepancy resolutions etc. These conditions might cause significant fluctuation in Token pricing.

Smart contract limitations

Smart contract technology is still in its early stages of development, and its application is of experimental nature. This may carry significant operational, technological, regulatory, reputational and financial risks. Consequently, although the audit conducted by independent third party increases the level of security, reliability, and accuracy, this audit cannot serve as any form of warranty, including any expressed or implied warranty that the 1WO Smart Contract is fit for purpose or that it contains no flaws, vulnerabilities or issues which could cause technical problems or the complete loss of 1WO Tokens.

Regulatory risks

The Blockchain technology, including but not limited to the issue of tokens, may be a new concept in some jurisdictions, which may then apply existing regulations or introduce new regulations regarding Blockchain technology-based applications, and such regulations may conflict with the current 1WO Smart Contract setup. This may result in substantial modifications of the 1WO Smart Contract, including but not limited to its termination and the loss of 1WO Tokens.

Price of Bitcoin and Ethereum

1World offers services to companies and individuals engaged in mining cryptocurrencies, primarily Bitcoin. Such operations are highly dependent on Bitcoin prices at local exchanges. Sharp and protracted decline in Bitcoin prices can affect the ability of 1World's customers to fulfill their contractual obligations to pay rental fees to token holders whose tokens they rent.

Rapid changes in technology may adversely affect mining business

Cryptocurrency mining is a very dynamic and fast-paced business. To remain competitive, 1World will use its best efforts to follow and promptly introduce the latest technologies at its facility. However, 1World's failure to remain competitive despite its endeavors may pose the risk of declining benefits for the 1WO Token holders. Likewise, token holders are advised to monitor their own mining equipment performance and update it as needed. Alternatively, as their equipment performance weakens over time, they should consider renting their tokens out to other miners to avoid the decline in the mining rewards.

Fluctuation in token benefits

The 1WO Token is intended to provide a valuable benefit of access to 1World ecosystem for cryptocurrency miners by giving them the ability to use tokens earning mechanism. Market changes, a drop in engagement value for publishers and / or reduction of traditional online display advertising may reduce the value of the 1WO Tokens and drive down the prices of tokens.

Sales and other taxes

Token holders and purchasers of mining equipment may be required to pay sales tax (collected at sale) and other taxes associated with the transactions contemplated herein, whether in the United States or in their home countries. It will be a sole responsibility of the token holders and purchasers of the mining equipment to comply with the tax laws of the United States and other jurisdictions and pay all relevant taxes.

Force Majeure

1World's performance may be interrupted, suspended or delayed due to force majeure circumstances. For the purposes of this White Paper, force majeure shall mean extraordinary events and circumstances which could not be prevented by 1World and shall include: acts of nature, wars, armed conflicts, mass civil disorders, industrial actions, epidemics, lockouts, slowdowns, prolonged shortage or other failures of energy supplies or communication service, acts of municipal, state or federal governmental agencies, other circumstances beyond 1World's control, which were not in existence at the time of Token Launch. If such circumstances occur prior to issuance of 1WO tokens and 1World is unable to issue 1WO Tokens within 6 months from the projected date, the escrow agent may issue a refund at the request of the 1WO Token purchasers. The refund will be issued in the original form of payment at the exchange rate on the date of the refund.

Compliance with U.S. laws and regulations

Because the hosting facilities are located in the United States, 1WO Token holders who wish to use their tokens to host their equipment at the facilities would be required to comply with the U.S. laws and regulations and may need to verify their identities and provide proof of address (for individuals), or verify their registration, good standing, list of ultimate beneficial owners, and address (for legal entities) prior to using their 1WO Tokens and setting up their equipment at 1World's facilities, or at any time thereafter upon 1World's request. Token holders who fail to comply with such verification request, or who are determined to be restricted from dealing with the U.S. entities or operating in the U.S., or who are otherwise ineligible under the US law to host their equipment with 1World would be refused hosting or 1WO Token rental services, with no refund issued by 1World for the purchased tokens. Such token holders may retain their tokens or may, at their discretion, choose to sell them to eligible customers. Token purchasers are solely responsible for learning about the US laws and legal restrictions applicable to residents of certain countries and individuals involved in certain activities.

Disclosure of information

Personal information received from 1WO Token holders, 1WO Token renters, and owners of the equipment submitted for hosting, the information about the number of tokens or miners

serviced by 1World, rewards earned on the pool, the wallet addresses used, and any other relevant information may be disclosed to law enforcement, government officials, and other third parties when 1World is required to disclose such information by law, subpoena, or court order. 1World shall at no time be held responsible for such information disclosure.

Value of 1WO Token

Once purchased, the value of 1WO Tokens may significantly fluctuate due to various reasons. 1World does not guarantee any specific value of the 1WO Token over any specific period of time. 1World shall not be held responsible for any change in the value of 1WO Token. Assumptions with respect to the foregoing involve, among other things, judgments about the future economic, competitive and market conditions and business decisions, most of which are beyond the control of the 1World project team and therefore difficult or impossible to accurately predict. Although the 1World team believes that its assumptions underlying its forward-looking statements are reasonable, any of these may prove to be inaccurate. As a result, the 1World team can offer no assurances that the forward-looking statements contained in this White Paper will prove to be accurate. In light of the significant uncertainties inherent in the forward-looking statements contained herein, the inclusion of such information may not be interpreted as a warranty on the part of 1World or any other entity that the objectives and plans of the 1World project will be successfully achieved.

Other risks

Please note that the 1World project may be subject to other risks not foreseen by its management at this time.

References

- [1] Naval Ravikant - a Conversation with Ryan Shea @ Blockstack Summit 2017
<https://youtu.be/IrSn3zx2GbM>
- [2] Digital Advertising Spending by Dentsu Aegis
http://www.dentsuaegisnetwork.com/media/dentsuaegisnetworknewsdetaila/2017/2017_06_15?Global-ad-spend-to-hit-5634-billion-in-2017-with-digital-driving-growth
- [3] Plasma.io - White paper: <http://plasma.io>
- [4] Raiden Technical paper: <http://raiden.network>
- [5] Lightning Network: <https://lightning.network>
- [6] Examples of 1World Platform deployment: <https://ico.1worldonline.com/examples>
- [7] Blockchain Law Group: <http://blockchainlawgroup.com/#services>

Disclaimer

Nothing provided herein shall be deemed to constitute a prospectus of any sort or a solicitation for investment, nor does it in any way pertain to an offering or a solicitation of an offer to buy any securities in any jurisdiction. This document is not created in accordance with, and is not subject to, laws or regulations of any jurisdiction which are designed to protect investors.

An information provided herein does not constitute an offer, solicitation or sale of the 1WO tokens in any jurisdiction in which such offer, solicitation or sale would be unlawful. Some restrictions on purchase of 1WO tokens may apply. Please check with your legal adviser.

Certain statements and estimates contained herein constitute forward-looking statements or information. Such forward-looking statements or information involve known and unknown risks and uncertainties which may cause actual events or results to differ materially from the estimates or the results implied or expressed in such forward-looking statements.

The exchange rate and a discount are not guaranteed and may vary for different purchasers depending on the price of purchased 1WO Tokens, the total number of 1WO Tokens sold, and other factors. Exchangeability of 1WO Tokens is subject to the availability of other projects' tokens.

**1X2
coin**

WHITEPAPER

TABLE OF CONTENTS

- ▶ **Introduction**
- ▶ **Masternodes**
- ▶ **Coinholders**
- ▶ **Proof-of-stake**
- ▶ **Coin specification**
- ▶ **Pre-mine and project funding**
- ▶ **Reward table**
- ▶ **Roadmap**

Introduction

The total value of the global sports betting market is difficult to estimate because of lack of consistency in how it is regulated in some parts of the world. Sports betting makes 30 to 40 percent of global gambling market. Size of one part of global gambling (online) is estimated as 37.9bn USD in 2017(1), and its constantly rising. There is also a trend inside global gambling that people are more and more moving from mainstream gambling (lotteries, casinos, poker and other gaming) to sports betting because of simple fact that there are more options and play look more fair.

In last few years, sports betting is developing in crypto space, and there are more and more sportsbook sites accepting BTC and other crypto currencies every month. In time this whitepaper is written there are more than 50 online sportsbooks accepting crypto.

The purpose of 1X2 Coin is to become one of crypto currencies accepted in crypto sportsbooks as payment option (deposit/withdraw), and that's just 1st phase. We will affiliate all good sportsbooks in market and future plan is to develop our own crypto enabled sportsbook site which will enter strong in this multibillion market.

We will start as advanced Masternode coin with progressive reward scheme and very attractive ROI for our investors.

(1) <https://www.statista.com/topics/1740/sports-betting/>

Masternodes

Constant growth and the innovation of our systems is a crucial objective for our strategy. Therefore, 1X2 is a self-funded system that has its own dedicated masternodes, used as budget to maintain and improve the project.

Based on this model we have developed financial structure that would generate income for 1X2 Coin owners while utilizing 1X2 Coin itself for security of the blockchain. In masternodes networks users of the platform and owner of 1X2 Coin are compensated by the network through allocation of rewards based upon 1X2 Coin owner contributions to the network as confirmation nodes and masternodes.

1X2 Coin is distributed within a hybrid network for securing the blockchain by confirming transactions while ensuring the privacy of transactions and facilitating instant transaction with users of our platform. 1X2 Coin is not limited to securing the network, it can be used as instrument for payments.

The main goal of 1X2 Coin is to serve as the cryptocurrency that is used by our partners, worldwide sportsbooks which accept cryptocurrency, for transfer of funds and payments within our platform.

Coinholders

Coinholder or Staker is an individual who owns some type of digital asset such as cryptocurrency and one individual can be a stakeholder of many different types of cryptocurrencies. Most cryptocurrencies provide their own wallets via link on their websites. Coins can then be transferred from the exchange into the wallet for storage and distribution purposes.

A Coinholder or staker, collect passive rewards on any amount of 1X2 Coin in their wallet. When visiting any affiliate exchange users can purchase 1X2 Coins and purchased coins can be stored in their digital wallet secured by the password. Here coins "stake" and verify other parties' transactions with purpose to earn more coins for the holder. By using our digital wallet users will be part of our exclusive coinholder network which is very user-friendly, so users can complete transfer of fund in quick and easy manner.

In first iteration of 1X2 Coin, coinholders collect all the benefits of 1X2 Coin's digital currency without ability to vote on the blockchain for the initiatives pursues.

Proof of stake

In traditional blockchain technologies, such as Bitcoin, preliminary proof-of-work concept was an energy-intensive blockchain technology based on the hashing protocol where miners are used to verify transaction on the network. This concept used substantial amount of energy, computation power and time needed to mine the coin. In this concept there is clear distinction between stakeholders and miners. Stakeholders are individuals who use network to facilitate a transaction. For example, if someone sends to someone one Bitcoin on the network, the block transaction is then verified by third party miners. Miners produce a mathematical computation based on the difficulty set by the network's parameter. The first miner to solve mathematical equation is rewarded and this is announced on the network before the block is created to be solved.

Proof-of-stake algorithm is based on storage of all the operations in the 1X2 Coin wallet with the distributed database. This system is based on the principles of decentralized management in the absence of single controlling authority. Synchronization of 1X2 Coin nodes which is running on proof-of-stake is done through the peer-to-peer network, P2P. Proof-of-stake is more efficient and environmental friendlier than proof-of-work, which utilizes lots of energy by using specific integrated circuit (ASIC) machines. This method can be use without expensive mining equipment on a normal entry level computer. 1X2 Coin user-friendly proof-of-stake removes the need for miners altogether. With proof-of-stake users earn rewards on the coins held in their wallet by following few simple steps:

1. Download the 1X2 Coin wallet from website
2. Purchase 1X2 Coin from an affiliate exchange
3. Place coins in digital wallet
4. Coin earn rewards through being staked as collateral to verify transactions on the blockchain

Coin specification

Coin Name:	1X2 Coin
Ticker:	1X2
Algorithm (POW/POS):	X11/POS
Total Supply:	21000000 1X2
Pre-mine:	180000 1X2 (0.85%)
Type:	Proof of Stake
Masternode Collateral:	1000 1X2
Masternode reward:	80% - 95%
POS reward:	20% - 5%
Block reward:	1 - 12 1X2
Block time:	60 seconds

Reward Table

Below listed is the block reward distribution table for 1X2 Coin masternode owners and coinholders, those which have their wallets open for staking.

Block range	Block reward	%MN reward	%Stake reward	MN reward	Stake reward
0-15000	1	80	20	0.8	0.2
15001-22000	1.25	80	20	1	0.25
22001-29000	1.5	80.2	19.8	1.203	0.297
29001-36000	1.75	80.4	19.6	1.407	0.343
36001-43000	2	80.6	19.4	1.612	0.388
43001-50000	2.25	80.8	19.2	1.818	0.432
50001-57000	2.5	81	19	2.025	0.475
57001-64000	2.75	81.2	18.8	2.233	0.517
64001-71000	3	81.4	18.6	2.442	0.558
71001-78000	3.25	81.6	18.4	2.652	0.598
78001-85000	3.5	81.8	18.2	2.863	0.637
85001-92000	3.75	82	18	3.075	0.675
92001-99000	4	82.2	17.8	3.288	0.712
99001-106000	4.25	82.4	17.6	3.502	0.748
106001-113000	4.5	82.6	17.4	3.717	0.783
113001-120000	4.75	82.8	17.2	3.933	0.817
120001-127000	5	83	17	4.15	0.85
127001-134000	5.25	83.2	16.8	4.368	0.882
134001-141000	5.5	83.4	16.6	4.587	0.913
141001-148000	5.75	83.6	16.4	4.807	0.943
148001-155000	6	83.8	16.2	5.028	0.972
155001-162000	6.25	84	16	5.25	1
162001-169000	6.5	84.2	15.8	5.473	1.027
169001-176000	6.75	84.4	15.6	5.697	1.053
176001-183000	7	84.6	15.4	5.922	1.078
183001-190000	7.25	84.8	15.2	6.148	1.102
190001-197000	7.5	85	15	6.375	1.125
197001-204000	7.75	85.2	14.8	6.603	1.147
204001-211000	8	85.4	14.6	6.832	1.168
211001-218000	8.25	85.6	14.4	7.062	1.188
218001-225000	8.5	85.8	14.2	7.293	1.207
225001-232000	8.75	86	14	7.525	1.225
232001-239000	9	86.2	13.8	7.758	1.242
239001-246000	9.25	86.4	13.6	7.992	1.258
246001-253000	9.5	86.6	13.4	8.227	1.273
253001-260000	9.75	86.8	13.2	8.463	1.287

Block range	Block reward	%MN reward	%Stake reward	MN reward	Stake reward
260001-267000	10	87	13	8.7	1.3
267001-274000	10.25	87.2	12.8	8.938	1.312
274001-281000	10.5	87.4	12.6	9.177	1.323
281001-288000	10.75	87.6	12.4	9.417	1.333
288001-295000	11	87.8	12.2	9.658	1.342
295001-302000	11.25	88	12	9.9	1.35
302001-309000	11.5	88.2	11.8	10.143	1.357
309001-316000	11.75	88.4	11.6	10.387	1.363
316001-323000	12	88.6	11.4	10.632	1.368
323001-330000	11.75	88.8	11.2	10.434	1.316
330001-337000	11.5	89	11	10.235	1.265
337001-344000	11.25	89.2	10.8	10.035	1.215
344001-351000	11	89.4	10.6	9.834	1.166
351001-358000	10.75	89.6	10.4	9.632	1.118
358001-365000	10.5	89.8	10.2	9.429	1.071
365001-372000	10.25	90	10	9.225	1.025
372001-379000	10	90.2	9.8	9.02	0.98
379001-386000	9.75	90.4	9.6	8.814	0.936
386001-393000	9.5	90.6	9.4	8.607	0.893
393001-400000	9.25	90.8	9.2	8.399	0.851
400001-407000	9	91	9	8.19	0.81
407001-414000	8.75	91.2	8.8	7.98	0.77
414001-421000	8.5	91.4	8.6	7.769	0.731
421001-428000	8.25	91.6	8.4	7.557	0.693
428001-435000	8	91.8	8.2	7.344	0.656
435001-442000	7.75	92	8	7.13	0.62
442001-449000	7.5	92.2	7.8	6.915	0.585
449001-456000	7.25	92.4	7.6	6.699	0.551
456001-463000	7	92.6	7.4	6.482	0.518
463001-470000	6.75	92.8	7.2	6.264	0.486
470001-477000	6.5	93	7	6.045	0.455
477001-484000	6.25	93.2	6.8	5.825	0.425
484001-491000	6	93.4	6.6	5.604	0.396
491001-498000	5.75	93.6	6.4	5.382	0.368
498001-505000	5.5	93.8	6.2	5.159	0.341
505001-512000	5.25	94	6	4.935	0.315
512001-519000	5	94.2	5.8	4.71	0.29
519001-526000	4.75	94.4	5.6	4.484	0.266
526001-533000	4.5	94.6	5.4	4.257	0.243
533001-540000	4.25	94.8	5.2	4.029	0.221
540001-4850000	4	95	5	3.8	0.2

Pre-mine and project funding

The pre-mine of 1X2 Coin is 180000 coins,
0,85% of total supply and one of smallest in industry.

From those, 60000 coins (33.33%) will be offered to public presale. If any coin from presale is not sold, we will burn them 24h after presale and that will be announced properly. Funds from coin sale will be used for listing on initial 5 exchanges, listing on MN portals, bounties, paid news articles, YouTube channels, google ads and all sort of marketing activities during 12-month period until our own sportsbook is launched.

Roadmap

- PROJECT CREATION AND CORE TEAM FORMATION
- ASSEMBLING CORE TEAM MEMBERS
- OFFICIAL WEBSITE LAUNCH
- INITIAL WHITEPAPER LAUNCH
- 1X2 TESTNET LAUNCH
- BITCOINTALK ANN POST

Phase 1

Oct18



- CONTACTING ALL 30+ CRYPTO ONLY SPORTSBOOKS
- SETTING UP AFFILIATE NETWORK
- OFFICIAL TWITTER, TELEGRAM, DISCORD LAUNCH
- GITHUB LAUNCH
- WALLETS FOR WINDOWS, LINUX, MAC uploaded on GITHUB
- PRESALE

Phase 2

Oct18



- MASTERNODES.ONLINE LISTING + BANNER
- BOUNTY LAUNCH
- CRYPTO-BRIDGE EXCHANGE LISTING
- COINEXCHANGE LISTING
- COINMARKETCAP LISTING
- CRYPTOPIA EXCHANGE LISTING + FEATURED CURRENCY&TIPSLIST

Phase 3

Nov18



- SHARED MASTERNODE/POS SERVICES PARTENRSHIP
- MASTERNODE ORIENTED SITES LISTINGS
- EXTENDED MARKETING CAMPAIGN
- CONNECTING CRYPTO ONLY SPORTBOOKS TO OUR COMMUNITY
- GIVEAWAY REAL CASH BETTING SLIPS FOR OUR MEMBERS ON DAILY BASE
- SETTING UP TIPS AND LINES CHANNELS

Phase 4

Nov18-Dec18



- SETTING UP PAID VIP TIPS ON TELEGRAM
- DEPLOYING ANDROID APP WITH VIP TIPS AND 1X2 MN TRACKING
- ENTERING ONE OR MORE CRYPTO ONLY SPORTBOOK AS PAYMENT OPTION
- DEPLOYING OUR OWN CRYPTO ONLY SPORTSBOOK ACCEPTING BTC AND 1X2 COINS
- LISTING ON HIGHER VOLUME EXCHANGE
- SPECIAL REWARDS FOR OUR INITIAL INVESTORS

Phase 5

2019/2020



INTRODUCING **THE KWATT COIN**

~ TOKENIZED ELECTRICITY ~

POWERED BY



4NEW

POWER TO THE PEOPLE
LITERALLY





DISCLAIMER

~ ~ ~

This document and any other 4NEW documents do not constitute a prospectus of any sort and are not a solicitation for investment. The KWATT Coin does not represent an ownership or share in ANY public or private corporation, or other entity in any jurisdiction. The KWATT Coin is a coin that can be used to purchase goods and services within the 4NEW ecosystem.

Acquisitions of KWATT Coins through the initial coin offering are non-refundable. KWATT Coins are only to be used in connection with 4NEW. Any acquisition and use of KWATT Coins carries significant financial risk, including the use of experimental software.

Except where specifically indicated, the statements and information set forth in this Whitepaper are not intended to recite current or historical facts, and constitute forward-looking statements. Forward-looking statements may include the words "may," "will," "could," "should," "would," "believe," "expect," "anticipate," "estimate," "intend," "plan" or other words or expressions of similar meaning. These forward-looking statements are based on the current beliefs, plans, objectives, goals, expectations, anticipations and/or intentions of 4NEW with respect to future events. Although 4NEW believes that the expectations reflected in the forward-looking statements are reasonable, 4NEW cannot guarantee the successful establishment or operation of its systems and business or any future results, level of activity, performance or achievements.

Many factors discussed in this Whitepaper or otherwise affecting the matters discussed herein, some or all of which may be currently unknown to 4NEW or beyond 4NEW's control, will be important in determining the ability of 4NEW to establish and operate its systems and business. Consequently, actual results may differ materially from those that might be anticipated from the statements and information set forth herein. In light of these and other uncertainties, the statements and information set forth in this Whitepaper are for informational purposes only, should not be relied upon in making any purchase or other decision, are subject to change, and are not intended to establish or indicate any representation, warranty, commitment, undertaking, promise or contract made on the part of 4NEW to any person. 4NEW does not undertake any obligation to publicly update any forward-looking statements, whether as a result of new information, future events or otherwise, except as required by law. ADDITIONAL RISKS HIGHLIGHTED ON THE WEBSITE.



Ladies and Gentlemen,

The team at 4NEW is proud to announce the world's first ever coin that embodies electricity. Our product is grounded in necessities, solving two global & social problems; waste surplus and energy shortfall.

Our blockchain platform will be built on top of the underlying waste treatment infrastructure covering the entire supply chain from collection of waste to generation of electricity to application of it within the cryptocurrency transactions processing.

Given the utilitarian nature of our services, it is our belief that 4NEW will successfully integrate the blockchain network within the real world applications of energy consumption by the crypto community leading to widespread mainstream adoption.

Our seasoned management team, with over 300 years of collective experience, brings a vast and diverse perspective that has enabled 4NEW to explore rare and unique opportunities. We are excited to present a solution such as ours that will revolutionize and standardize four industries, Crypto-mining, blockchains, Waste Management and Energy, creating disruptive economies of scale on a global level.

Regards ,

Sandeep Golechha

Chief Executive Officer



PROBLEM

Cryptocurrencies global market capitalization has surpassed \$500 billion USD rising over 2000% in 2017. This trend is expected to continue for the foreseeable future as businesses increasingly embrace the elegant design and transparency the blockchain offers to all.

However, as of December 2017, Bitcoin mining energy intake has officially surpassed the entire energy consumption of Denmark. As the difficulty of mining increases to reflect the influx of miners joining the network, this energy consumption will increase.

The most astounding aspect of this is that the rate of expansion is exponential. At the current rate of consumption, next year Bitcoin mining will consume enough energy to be listed as the twentieth country in the world by energy consumption. The model is simply unsustainable. The world relies primarily on the production of energy from the burning of coal and oil, which not only damages the environment, but the economy as a whole. If Bitcoin has a great enough impact on the world's coal and oil supplies, the cost of a kilowatt will rise globally.

The more valuable one bitcoin becomes, the more energy will be used to mine that coin, therefore with price spikes, come energy spikes. This will go on until energy around the world will cost much more than it does currently, as a result of increased demand from miners globally.



KEY NETWORK STATISTICS¹

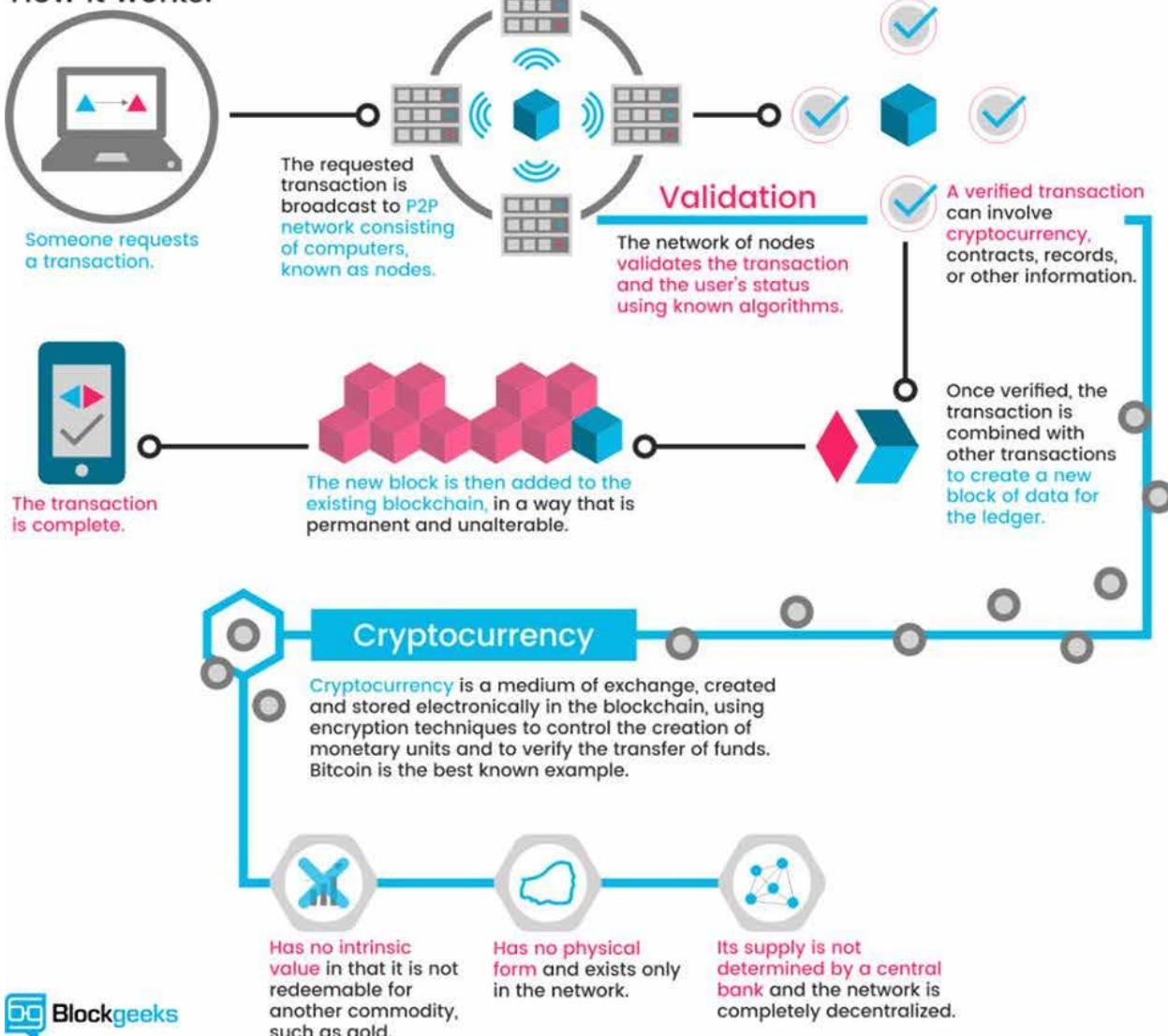
KEY NETWORK STATISTICS	BITCOIN	ETHEREUM
Network's current estimated annual electricity consumption* (TWh)	37.02	10.55
Annualized global mining revenues	\$19,107,870,821	\$7,125,662,989
Annualized estimated global mining costs	\$1,850,968,079	\$1,265,551,175
Country closest to in terms of electricity Consumption	Qatar	Georgia
Electricity consumed per transaction (KWh)	262	33.00
Number of U.S. households that could be powered in a year	3,427,719	976,506
Number of U.S. households powered for 1 day by the electricity consumed for a single transaction	8.85	1.12
Bitcoin's electricity consumption as a percentage of the world's electricity consumption	0.17%	0.05%
Annual carbon footprint (kt of CO2)	18,139	-
Carbon footprint per transaction (kg of CO2)	128.35	-

NOTE¹ - Data as of December 25, 2017. Data provided by Digiconomist Energy Consumption Index. <https://digiconomist.net>



A CRYPTOCURRENCY TRANSACTION

How it works:





4NEW SOLUTION

4NEW is the world's first eco-friendly, tangible, waste to energy blockchain solution. The concept is quite simple, the process of refining waste product into water and organic materials creates energy and that is then leveraged to either be sold to the national grid or applied to operate mining processes at its onsite mining farm.

The cost to produce the energy is met through the revenue generated from the waste collection services and sale of byproducts facilitating a sustainable operation at breakeven or a marginal profit. Therefore, the energy produced is unencumbered and freely available for utilization orsale to the national grid.

Historically, the price of 1 kilowatt has been very stable for the past fifty years at approximately \$0.15 globally, inflation adjusted. This trend is expected to continue for the foreseeable future in lieu of technological innovations. However, the wild card that no one has truly evaluated is the exponential acceptance of the blockchain worldwide leading to a massive spike in energy consumption by cryptocurrency mining that could drive the price of energy up globally. 4NEW has the unique opportunity to apply this finite lifetime supply of energy to its coin, namely, KWATT. The 4NEW coin symbol is KWATT². Each KWATT Coin embodies within it, 1 kilowatt of electricity for a year.

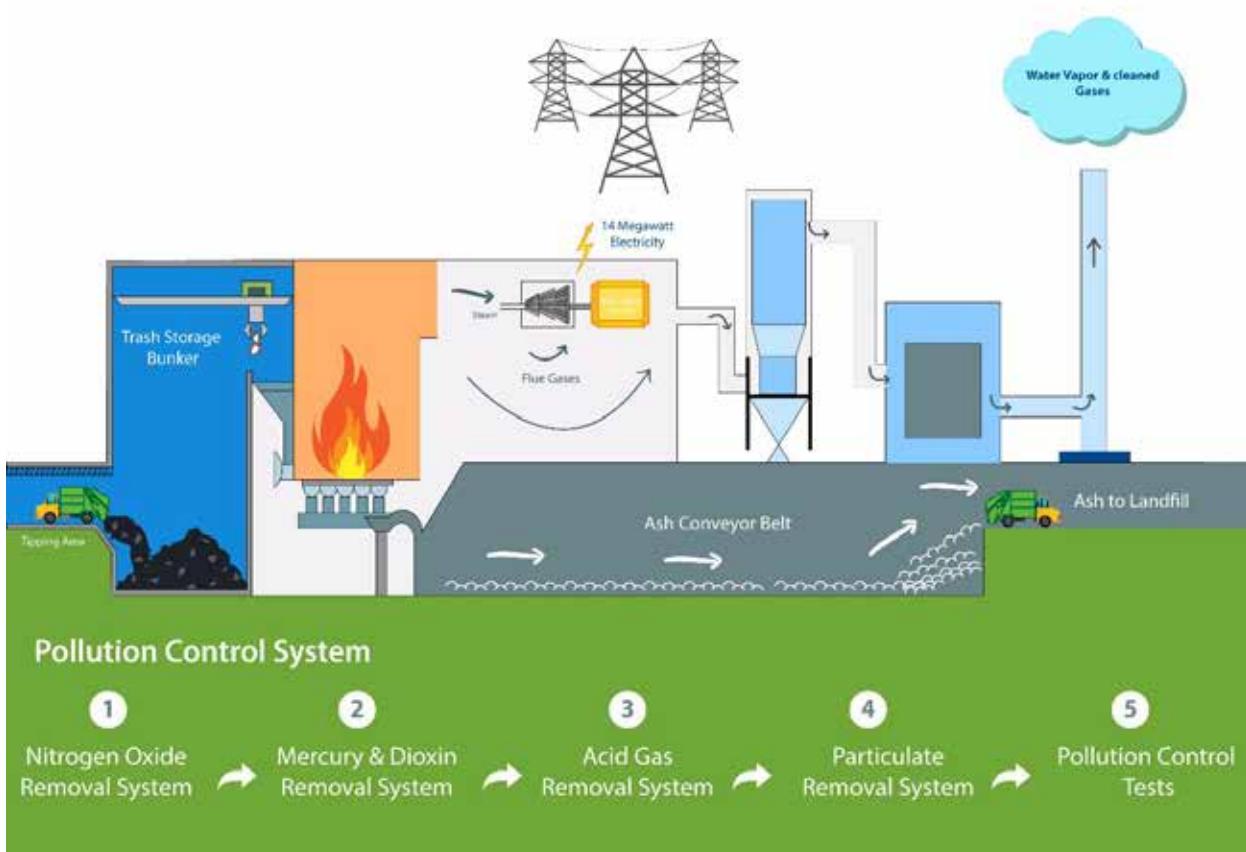
NOTE² - KWATT Coin was formerly named FRNCoin. It is the same coin, just a different coin symbol.



4NEW

POWER TO THE PEOPLE, LITERALLY.

4NEW'S WTE PLANT PROCESS





4NEW'S BLOCKCHAIN

4NEW is a producer of energy. Consequently, 4NEW enjoys a unique vantage point to apply its energy at a fixed price, inflation adjusted for a significant period of time. This will enable standardization of transaction fees within the crypto community.

This standardization of fees cannot be matched by any other blockchain as they have to purchase their energy from retailers.

Our blockchain will enable payment system platform that can accommodate all utilization of the KWATT Coin in varied and diverse industries.

The 4NEW decentralized, distributed ledger is where all actors in any industry will be able to transact using the KWATT coin. The coins are smart contracts which establish a binding relationship between transacting parties and provide a value for each transaction.

The ledger will provide an immutable and audit-able journal of all transactions related to purchase and sale of goods and services on the blockchain. With all parties to each transaction being able to see the same ledger entry, costs of reconciliation and potential issue of disputes and revenue leakage are controlled to a very large extent. This will standardize three industries of Waste, Energy and cryptocurrencies that are yearning for technological innovation. This will also enable KWATT.



4NEW'S KWATT TOKENOMICS

- KWATT Token currently is an ERC20, Ethereum based smart contract. Upon completion of the blockchain development, the token will be swapped to the KWATT Coin that will interact with our blockchain.
 - The total coin offering is for three hundred million coins (300,000,000). This is a hard cap limit for the crowd sale offering.
 - Our first plant will launch with a capacity of generating 10 megawatts of power every hour. Upon seasoning the plant operations, our infrastructure will be able to increase output capacity to 40 megawatts per hour. 1 megawatt is equivalent to 1000 kilowatts. 1000 kilowatts powers 1000 households for one day. Peak or off-peak usage of the power at different times of the day can cause this average to deviate.
 - The maximum annual output capacity of the plant is 346 million kilowatts per year. Due to maintenance and general down time for repairs to the plant, expected annual output capacity is estimated at 300 million kilowatts per year realistically.
 - Each KWATT Coin embodies an annual supply of 1 kilowatt of electricity within it.
 - A typical Waste to Energy plant depreciates to its salvage value over 50 years. Regular maintenance and upkeep will allow us to extend life beyond that.
 - This means holder of KWATT Coin will be able to apply their energy to one of two places each year for the next 50 years. They can either sell their energy to the UK National Grid or they can choose to apply it towards 4NEW's cryptocurrency mining farm.
 - The price of 1 kilowatt for electricity is a very stable metric. Over the past 50 years, the global average retail price is approximately \$0.15 USD per kilowatt, inflation adjusted.
 - 4NEW will never authorize any additional coins issuance over and above the three hundred millions coins being launched in this initial coin offering. Therefore, any future growth in 4NEW plant sites will always rely on the supply of the coins being issued in this offering.
-



- Each year management will apply 35% of its net profits towards a reinvestment strategy to enable future development of plants. This will ensure longevity and scalability to 4NEW over a sustained period of time.
- 4NEW Insiders and Founders will be restricted from selling any coins until January 1st, 2019.
- Any KWATT Coins not sold in the offering will be burned. For the avoidance of doubt, all burned coins will release the supply of the energy that was embodied within the coin, allowing that unencumbered energy to be freely sold to the UK national grid or applied towards the mining farm at management's discretion.
- At the start of each year, KWATT Coin holders will be able to choose a desired application of their energy the coin holder owns as represented by the total amount of KWATT Coins in their control at the time of this election. Therefore, if the coin holders desire to sell their energy to the UK national grid then the respective option can be selected. Alternatively, if the coin holder were to select the mining farm then the energy will be applied to the mining farm.

Any decisions not made within the allotted time frame at the start of each year, will leave the management the right to determine the allocation of the energy at its discretion.

- Management, at its sole discretion, may decide to extend the ICO ending date to an uncertain end date.
- This document and any other 4NEW documents do not constitute a prospectus of any sort and are not a solicitation for investment. The KWATT Coin does not represent an ownership or share in ANY public or private corporation, or other entity in any jurisdiction. Acquisitions of 4NEW through the initial coin offering are non-refundable. KWATT Coins are only to be used in connection with 4NEW goods and services within its ecosystem only. Any acquisition and use of KWATT Coins carries significant financial risk, including the use of experimental software.



4NEW'S KWATT COIN FEASIBILITY

The KWATT Coin will represent a certain hashing capacity per coin. This concept is not new; Companies such as Giga Watt have offered similar mining items for lower costs, however, KWATT Coin is extremely unique in scope. We do not charge energy fees for mining, the only cost to a coin holder is the cost of the coin. This means that a coin holder will be able to mine all cryptocurrencies for the lifetime without spending an additional penny for their energy bill. The energy is free to us, so it is free also to the coin holders.

How does 4NEW Sustain Itself?

The concept of 4NEW relies upon the waste to energy model. In this model we are paid for the waste that we process, and the sale of byproducts such as fertilizer, organic materials and clean water. The start up costs to this mechanism are funded by the coin sale, and the plant's overhead is funded by cash flow generated from collection of waste and revenue from sale of byproducts. Additionally, 4NEW, and the 4NEW team will retain a portion of the KWATT Coins (and their associated mining capacity) which will provide an additional revenue stream moving forward.

The Mining Capacity of a KWATT Coin

The most difficult part of the KWATT Coin design has been determining a model to correlate with the increase in mining difficulty. We understand that one hash today can represent half of its mining power a month from now. To solve this issue, we have decided to have the coin represent a fraction of the total mining capacity of the 4NEW network rather than a fixed mathematical rate. This concept allows 4NEW to expand their mining capacity to match a competitive rate on the network. This rate of exponential expansion will be a predetermined reinvestment strategy of the funds received through 4NEW's own mining portfolio, in addition to the profits from the other revenue streams. This model not only guarantees the longevity of free energy, but the longevity of competitive mining practices.



Portfolio Customization

Users will have the ability to decide which coins or coins they would like to put their KWATT Coin power towards to mine. The options will consist of the top twenty minable coins, this decision will automatically point the necessary amount of hash rate towards mining that coin, and the yield will be transacted to the account associated with your 4NEW Wallet.

Proof-of-Work / Proof-of-Stake

In recent months, Ethereum has taken steps towards a Proof of Stake system that will be fully implemented sometime in the following years. At 4NEW we fully support these steps and understand that more efficient systems are necessary for the sustainability of cryptocurrency in the long term. Yet, we also understand that Proof of Work will not likely be fully removed from cryptocurrency within the next decade. For this reason, additional precautionary steps must be taken to reduce the economic and environmental effects of the inefficiencies associated with Proof-of-Work mining, and our mission is to be on the vanguard of these efforts. Even if Proof of Work was completely removed and Bitcoin mining non-existent, the energy embodied within the coin can still be either applied to the Proof of Stake mining operations or the national grid, given severe energy shortfalls already prevalent within the world.

Management and KWATT Coin Holder Interests aligned

Given that the Waste to Energy plant will sustain its operations at breakeven from revenue generated from the sale of waste collection services and byproducts, the energy produced is free. This lifetime supply of free energy is being purchased by the coin holder in this crowdsale. Any revenue generated from the administrative and facilitation fees the company will charge to either sell the energy to the national grid or apply it to the crypto-mining farm on behalf of the coin holders will allow for future growth and expansion strategy. Therefore, increasing the overall demand for the coin. With three plants, the total output capacity rises to roughly 1 billion kilowatts per annum. This will enable a market capitalization of the coin to rival most successful cryptocurrencies.



4NEW'S KWATT COIN ADDITIONAL UTILITY

Please note, the utilization of the KWATT Coin referenced below is in addition to the energy applications of the coin within the waste to energy and the crypto mining industry.

Our blockchain will be under development starting 2018 as per our roadmap. 4NEW has received significant interest from other institutions in diverse industries to integrate the KWATT Coins within their services.

Upon attaining critical mass with Business to Business adoption, KWATT is positioned to be the next Bitcoin. Starting 2018, KWATT Coin holders will be able to use their coins for the following services as a payment tender for respective services. Beta testing will initiate in 2018 for the following businesses:

- **Mining Farm** - 4NEW's mining farm will be onsite of the WTE plant. The KWATT Coin will interact with the miners to enable them access to the blockchain in order to mine currencies of their choosing.
- **Money Transfer company** - Two licensed operating global money transfer companies with a collective monthly money transfer turnover of over \$25 Million USD in 15 year track record. This coin utility will enable KWATT to operate as a store of value allowing individuals to transfer funds globally using the KWATT Coin as a vessel.
- **Licensed Pharmacy** - A licensed pharmacy with over USD \$24 Million in sales in the United States and 18 year track record. Upon completion of beta testing in March, individuals globally will be able to purchase their medications using the KWATT Coin from a fully licensed pharmacy online.
- **Insurance Company** - In 2018 we will initiate beta testing of the utilization of the KWATT Coin as a payment system for a 1.5 billion dollar healthcare insurance company with over 8 year track record. With over 90,000 members covered under their policies, anticipation is that the KWATT Coin will break critical mass and enter main stream utilization.

4NEW Management will continue to work with operators in various industries to facilitate widespread adoption and utilization of its coin through 2018.



THE 4NEW ROADMAP

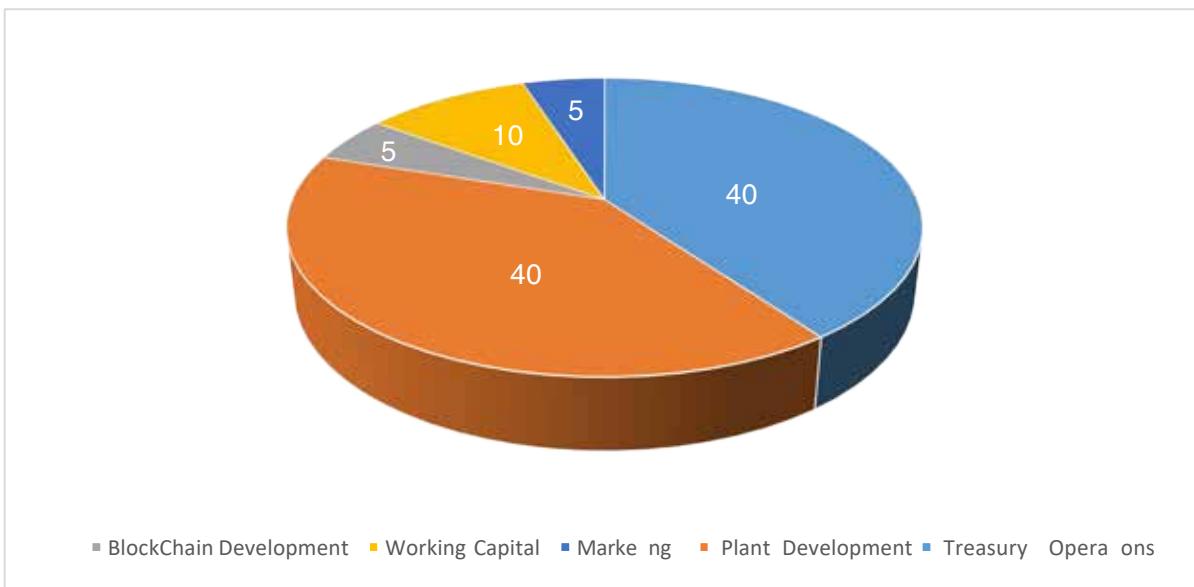
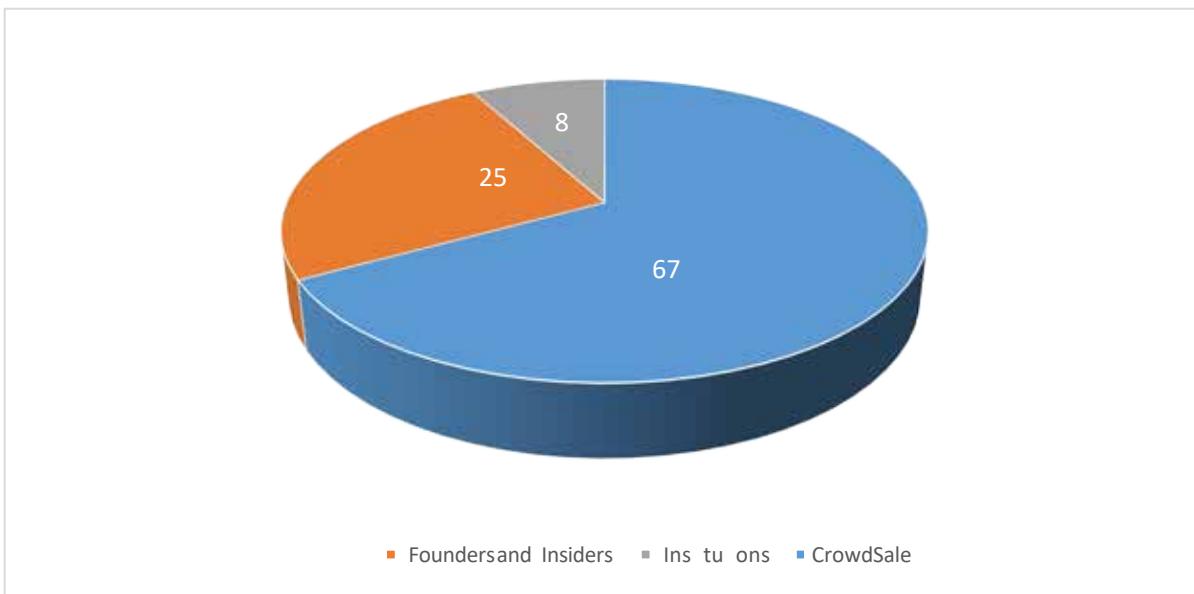
OCTOBER 17, 2017	Pre-Sale Round 1 Launch of the KWATT Coin
NOVEMBER 14, 2017	Closing the Pre-Sale Round 1
NOVEMBER 15, 2017	4NEW Pre-Sale Round 2
DECEMBER 15, 2017	Closing of Pre-Sale Round 2
JANUARY 1, 2018	4NEW Initial Coin Offering Live
JANUARY 31, 2018	4NEW Initial Coin Offering Closing
FIRST QUARTER 2018	<p>Complete purchase of 4NEW plant sites with planning permissions & regulatory licenses in place already identified:</p> <ul style="list-style-type: none">1. Sheffield, UK - 30 Acres2. Hertfordshire, UK - 10 Acres3. NewCastle, UK - 5 Acres- Initiate development of blockchain platform- Initiate 4NEW Smart Meter approval process with the UK SMETS 2 protocols- Beta Testing KWATT Coin utility with service providers
SECOND QUARTER 2018	<ul style="list-style-type: none">- Begin installation of Waste to Energy Plant on all three sites- Complete Blockchain platform integration with 4NEW Smart Meter- Market trial and testing of 4NEW Smart Meter system with Blockchain integration
THIRD QUARTER 2018	<ul style="list-style-type: none">- Marketing efforts to build community awareness of 4NEW Services- Develop and initiate relationships with key aggregators with access to wholesale quantity of waste- Stress test or beta test blockchain platform prior to going live
FOURTH QUARTER 2018	Installation of Waste to Energy Plants complete and facilities will be operational



4NEW

POWER TO THE PEOPLE, LITERALLY.

4NEW ALLOCATION DISTRIBUTION





THE KWATT COIN VALUATION

DISCOUNTED CASH FLOW ANALYSIS

ASSUMPTIONS:

DISCOUNT (INFLATION) RATE: 2%

EXPECTED ANNUAL CASH FLOW: \$0.20

NUMBER OF PAYMENTS: 50

$$NVP = -C_0 + \frac{C_1}{1+r} + \frac{C_2}{(1+r)^2} + \dots + \frac{C_T}{(1+r)^T}$$

$-C^0$ = Initial Investment

C = Cash Flow

r = Discount Rate

T = Time

NET PRESENT VALUE (NPV) = \$6.26

Please note, the Net Present Value calculated above is simply the present value of all future cash flow anticipated from the energy produced over the lifetime of the plant on a per kilowatt basis. This is not a representation of the value of utility coin. External influences may cause the coin to be priced higher or lower than these projections. Valuation analysis is not a guarantee of any returns or results. Also, FX conversion rates could cause the value of cash flow assumption to change with changes in the market.



THE KWATT COIN PRICING

Each KWATT Coin is equal to 1kW of electricity.

Global average retail price per kW of electricity is \$0.15.

Global average wholesale price per kW of electricity is \$0.05.

4NEW mining operation will pay \$0.20 per kW. Therefore, each coin represents an intrinsic value of \$0.20 per 1 kW of electricity.

All energy generated by the plant will be owned by the KWATT Coin holders.

KWATT Coin holder can either sell this power to the UK national grid or apply it to 4NEW's mining farm.

Net revenues, after administrative and facilitation fees, generated from the sale of the energy to the crypto mining farm will be distributed to the coin holder annually in cryptocurrencies.

Net revenues, after administrative and facilitation fees, generated from the sale of the energy to the UK national grid will be retained for future plant build out leading to additional KWATT Coins being delivered to the coin holder.

Revenue distribution will initiate 24 months after the completion of the Initial Coin Offering. The first 12 months will be plant buildout period and the second 12 months will be reserve build up period. Distribution will be annually thereafter.



MINING CASE STUDY³

BITMAIN CASE STUDY

The energy efficiency of Bitcoin mining is significantly worse than initially anticipated, and the associated carbon footprint is worrisome. One test case of cryptocurrency mining is Bitmain. Bitmain's mining operation consists of 25,000 units in total, seven buildings onsite that contain 21,000 ASIC Bitcoin rigs, and one building contains 4,000 Litecoin rigs all producing around \$200,000 profit daily after paying for 40MW of five cents per kilowatt-hour coal-powered electricity at around \$40,000, and 50 salaried employees.

One of the biggest liabilities of such a large mining operation is heat. As the temperature outside rises, the 5,000 BTU per hour producing mine begins to malfunction at a much higher frequency than typical. To counteract this heat, the hot air is pumped from the facility and the machines themselves are cooled with evaporative coolers. These systems collectively consume about one fifth of the total power consumption of the mining facility.

The total footprint of the mine is estimated to be around 24-40 tons of CO₂ per hour. Assuming an identical energy efficiency, which is quite generous, this means the total energy consumption of the Bitcoin network processes about 200 tons of CO₂ per hour. However, this footprint could be significantly reduced if the source of energy is substituted for a greener method.

This mining farm commands approximately 5% of the global bitcoin mining market share.

Ironically, Bitmain's plant consumes 40 megawatts per hour, which happens to be the output capacity of the 4NEW plant. Therefore, it would be reasonable to envision our operation similar in scale and scope to Bitmain's. This is why we felt it prudent to share this case study with you.

NOTE³ - Case Study conducted by Digiconomist. <https://digiconomist.net>



COMPLIANCE

ACCOUNTING

4NEW's accounting financial statements will be maintained at Zucker Forensics P.A. Zucker Forensics is a credentialed forensic accounting firm with accredited US Certified Public Accountants. With over 35 years of forensic accounting experience in identifying fraudulent accounting practices, the management at 4NEW deemed it necessary to retain such a capable team to manage its books and records.

INDEPENDENT AUDITOR

4NEW's independent auditor is Daszkal Bolton LLP. Founded over 26 years ago, Daszkal Bolton maintains an illustrious track record of thorough audits of high growth companies in every sector. In an effort to abide by the highest ethical standards, 4NEW Management will submit to annual audits to help secure our investor's trust and confidence.

ABLE TO THE WORLD

ABLE:
APPLYING BLOCKCHAIN TO EXTEND FINANCE



This White Paper states the current views of ABLE token issuer and service provider, Chain Holdings OÜ (registration code 14406869) and its Korean agency, K-Blockchain (registration code 462-86-00783) concerning the ABLE Platform and related matters. Chain Holdings OÜ or K-Blockchain may from time to time revise this White Paper in any respect without notice. The information presented in this WhitePaper is indicative only and is not legally binding on Chain Holdings OÜ or K-Blockchain or any other party. This document is for informational purposes only and does not constitute and is not intended to be an offer to sell, a solicitation of an offer to buy, or a recommendation of; (1) ABLE Tokens, (2) an investment in the ABLE Platform or any project or property of Chain Holdings OÜ or K-Blockchain, or (3) shares or other securities in Chain Holdings OÜ or K-Blockchain or any affiliated or associated company in any jurisdiction. Please read the important legal disclaimers at the end of this White Paper.

CONTENTS

0.VISION	4
1. INTRODUCTION	5
2. BACKGROUNDS	8
2.1 MARKET SIZES OF BANK LENDING-DEPOSIT SPREADS	
2.2 CASE STUDIES OF CRYPTOCURRENCY BANKING SERVICES	
3. ABLE PROJECT: FINANCIAL SERVICES	11
3.1 ABLE INVESTING-LENDING MATCHING ENGINE	
3.2 CRYPTOCURRENCY PAYROLL SERVICE AND CREDIT SCORE/ LOAN	
3.3 CONVENIENT CRYPTOCURRENCY ADDRESS AND SCHEDULED REMITTANCE SERVICE	
3.4 ACCOUNT-BASED ICO SMART CONTRACT	
4. ABLE ECOSYSTEM DEVELOPMENT PLAN	16
4.1 ABLE DEX	
4.2 ABLE FINANCIAL PRODUCTS	
4.3 MICROPAYMENT / THIRD PARTY	
5. ABLE SYSTEM ARCHITECTURE	19
5.1 ABLE SYSTEM	
5.2 ABLE ACCOUNT	
5.3 ABLE CRYPTOCURRENCY ECONOMIC SYSTEM	
5.4 ABLE CONSENSUS PROTOCOL	
6. TECHNICAL STRUCTURE OF ABLE SYSTEM	21
6.1. ARCHITECTURE OF ABLE SYSTEM	
6.2. APIs OF ABLE SYSTEM	
6.3. ABLE COIN ON ABLE SYSTEM	
7. BUSINESS MODEL	27
8. CONCLUSION	29

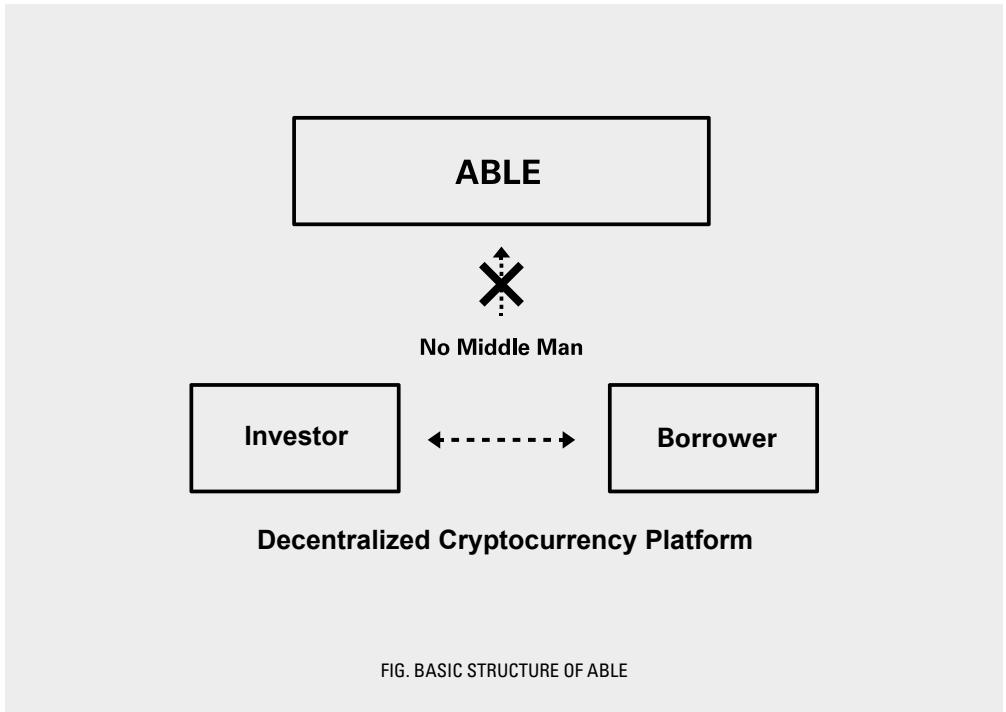
VISION

The ABLE project will create a solution to problems with traditional bank lending-deposit spreads, and nontransparent and centralized cryptocurrency financial services, basing the solution on the transparency and reliability of a blockchain system. The ABLE project is the cryptocurrency-specific financial solution that provides cryptocurrency financial products through person-to-person loan-matching systems and decentralized exchanges.

The ABLE project will create a platform that enables cryptocurrency-based financial activities to center around the ABLE account, emanating from the fact that banking activities are carried out primarily through bank accounts. Users propose loan interest rates through a matching system, eliminating the lending-deposit spread by directly connecting users on a peer-to-peer basis. The ABLE project supports decentralized exchanges and provides decentralized cryptocurrency wealth management services through smart contracts. Ultimately the project aims to evolve into the ABLE Ecosystem through its integration with external systems, including simple payments.

01 INTRODUCTION

01 INTRODUC- TION



ABLE PROJECT: DECENTRALIZED CRYPTOCURRENCY PLATFORM

The ABLE project has been proposed to provide an alternative to banks' conventional lending-deposit spread structure, and to solve problems with initial cryptocurrency banking services.

- 1) Banks have a business model based on a lending-deposit spread (i.e., the difference between lending and deposit interest rates), in which the bank acts as an intermediary between depositors and borrowers. This structure allows banks to receive deposits at low interest rates and lend money at high rates, profiting from the difference.
- 2) Existing cryptocurrency-based banking services are provided through a centralized structure. The combination of unsustainably high interest rates and centralized services renders banks directly vulnerable to managers' moral hazard and the hacking of central servers. The lack of integrated solutions in the crypto market withholds users from experiencing diverse banking services.

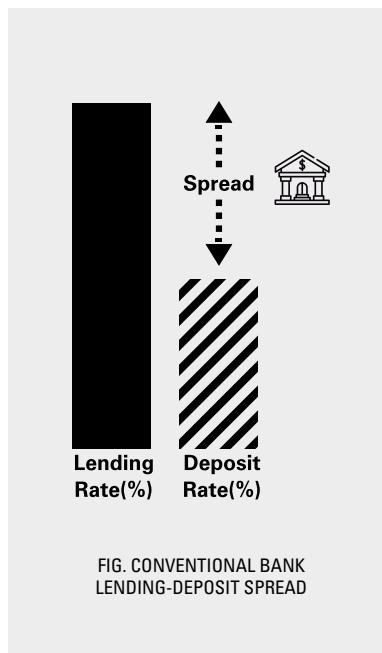


FIG. CONVENTIONAL BANK LENDING-DEPOSIT SPREAD

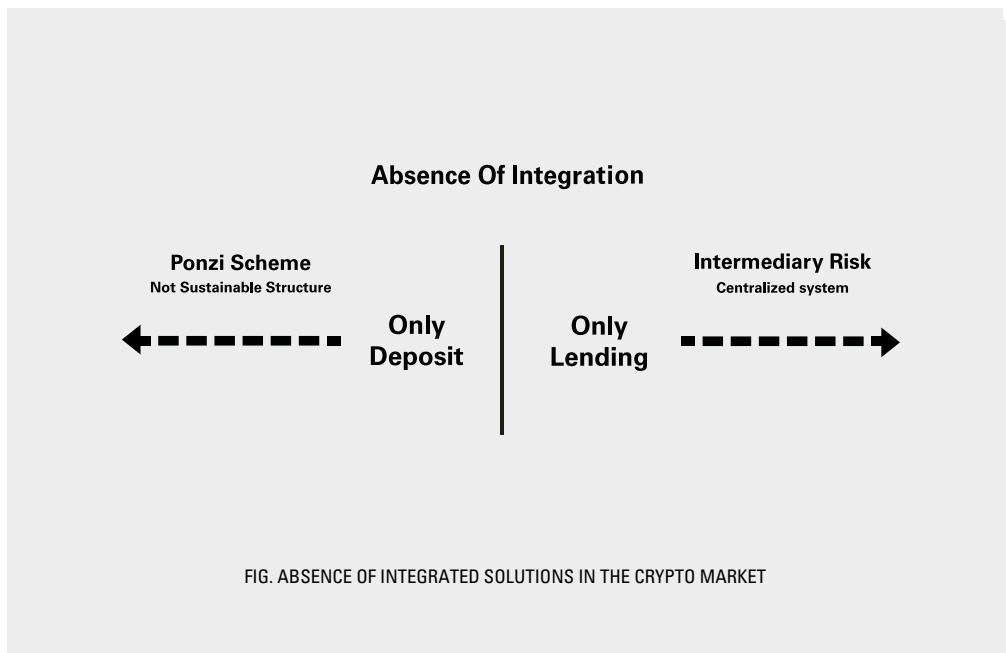


FIG. ABSENCE OF INTEGRATED SOLUTIONS IN THE CRYPTO MARKET

The ABLE project aims to solve these problems by applying decentralized blockchain technology.

By developing a matching engine that connects investors and borrowers on a peer-to-peer basis, the project directly connects investors and borrowers without any intermediaries. With intermediaries and lending spreads eliminated, cryptocurrency interest rates will be determined by mutual agreement between users in a free market, and said rates will serve as market interest rates. Investors can receive higher interests than they would in a centralized market, while borrowers can lend money at lower rates, creating a win-win scenario. Since investing and lending occur governed by smart contracts, the project will eliminate exposure to managers' moral hazard and the risk of central-server hacking.

Current cryptocurrency banking services provide deposits and loans separately rather than linking depositors and borrowers. The services have shortcomings such as inability to provide high deposit interest rates in a sustainable way. Such services cannot mitigate or eliminate intermediary risks because their structure is based on a centralized model. On the other hand, the ABLE project directly links users, primarily through an account, and enables a variety of financial activities such as payroll and investment, based on smart contracts.

The ABLE project primarily aims to apply decentralized blockchain technology to financial and wealth management features; to directly link demand and supply through an account on a peer-to-peer basis; and to establish a reliable system. It aims subsequently to build an in-house decentralized exchange, and then to evolve into a platform on which to develop and use smart contracts for finance and wealth management services.

02 BACKGROUNDS

- 2.1 MARKET SIZES OF BANK LENDING-DEPOSIT SPREADS
- 2.2 CASE STUDIES OF CRYPTOCURRENCY BANKING SERVICES

02

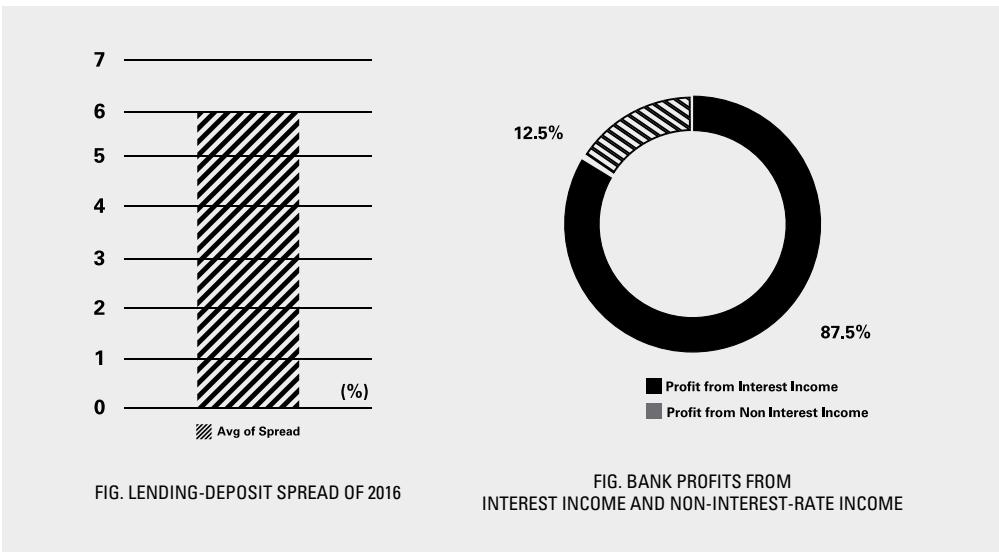
BACK-GROUNDS

TRADITIONAL BANK LENDING-DEPOSIT SPREAD AND CRYPTOCURRENCY BANKING BUSINESS MODEL

2.1 MARKET SIZE OF BANK LENDING-DEPOSIT SPREADS

Traditional banks serve as intermediaries and profit from differences between deposit and loan markets (lending-deposit spreads). The ABLE project will create a system that enables investors and borrowers to share the value generated by eliminating such intermediaries.

Currently, cryptocurrency banking businesses provide piecemeal services such as high-yield deposits and loans involving intermediary risks. The ABLE project creates a sustainable business model that enables direct account-based linking of users through smart contracts, and through decentralization solves the problems of managers' moral hazard and the hacking of central servers.

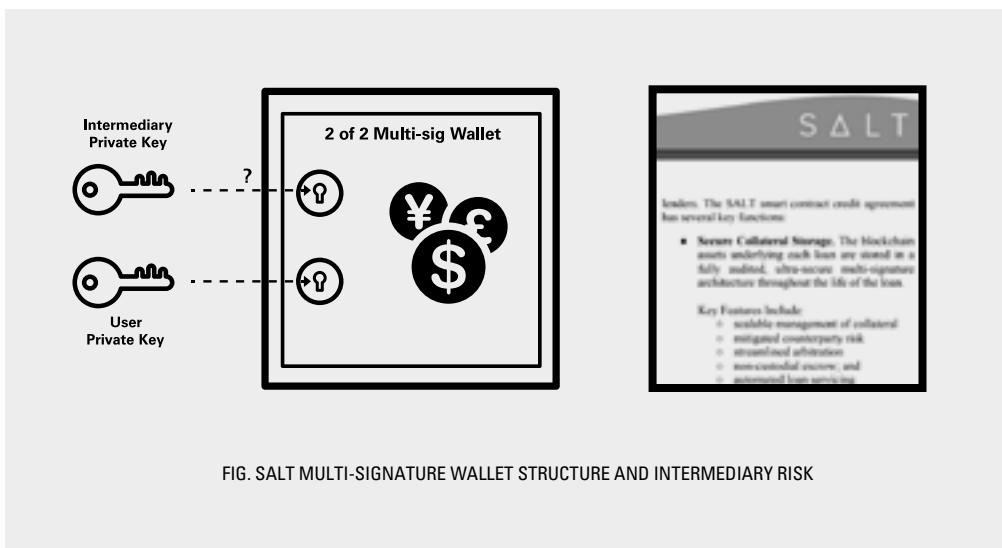


As of 2017, the average deposit-lending spread of banks was recorded at about 6%, thus explaining the 80% of their gross income. The non-interest income (e.g., service charges, trusts, securities, foreign-exchange income), on the other hand held a small proportion of 20%.

The decentralization of financial businesses eliminates operational risks that occur during bank operations. Since investors are directly linked to borrowers without banks as their intermediaries, they can use services without being burdened by banks' operational risks.

2.2 CASE STUDIES OF CRYPTOCURRENCY BANKING SERVICES

2.2.1 CASE STUDY OF LOAN BROKER RISK: SALT



SALT is a banking service cryptocurrency solution that lends money in fiat currency backed by cryptocurrency as collateral. Using the multi-signature technology that only allows a withdrawal with the signatures of both intermediaries and borrowers, this solution alleviates intermediary risk and thus distinguishes itself from existing solutions. However, the SALT structure stipulates that without signatures, users cannot receive ownership of cryptocurrencies in return. While the platform uses smart contracts, until it becomes decentralized it cannot be an ideal solution to intermediary problems. Most lending platforms are also vulnerable to managers' moral hazard because intermediaries retain customers' cryptocurrency assets.

2.2.2 PONZI-SCHEME TYPE CRYPTOCURRENCY DEPOSITS

There are cryptocurrency deposit services in the form of Ponzi schemes that pay unsustainably high deposit interest rates. These services attracted customers by providing interest rates as high as 10% per month in the six months following launch. The Government of the United Kingdom has imposed sanctions on Ponzi-type fraudulent cryptocurrency financial services. Ponzi-scheme type cryptocurrency services that collect money on an arbitrage profit-sharing model also have emerged.

2.2.3 CONCLUSION FROM ANALYSIS OF CASE STUDIES OF CRYPTOCURRENCY BANKING SERVICES

Cryptocurrency banking services are still in their infancy and do not have established business models. Most attempts to solve their problems have separated deposits from loans. This mechanism was not sustainable under the weight of high deposit rates and failed to solve the intermediary-risk problem caused by centralization.

The ABLE project aims to create a solution to existing problems and a better system through the investing-lending matching engine and various financial services that provide account-based links between investors and borrowers.

03 ABLE PROJECT : FINANCIAL SERVICES

3.1 ABLE INVESTING-LENDING MATCHING ENGINE

3.2 CRYPTOCURRENCY PAYROLL SERVICE AND CREDIT SCORE/ LOAN

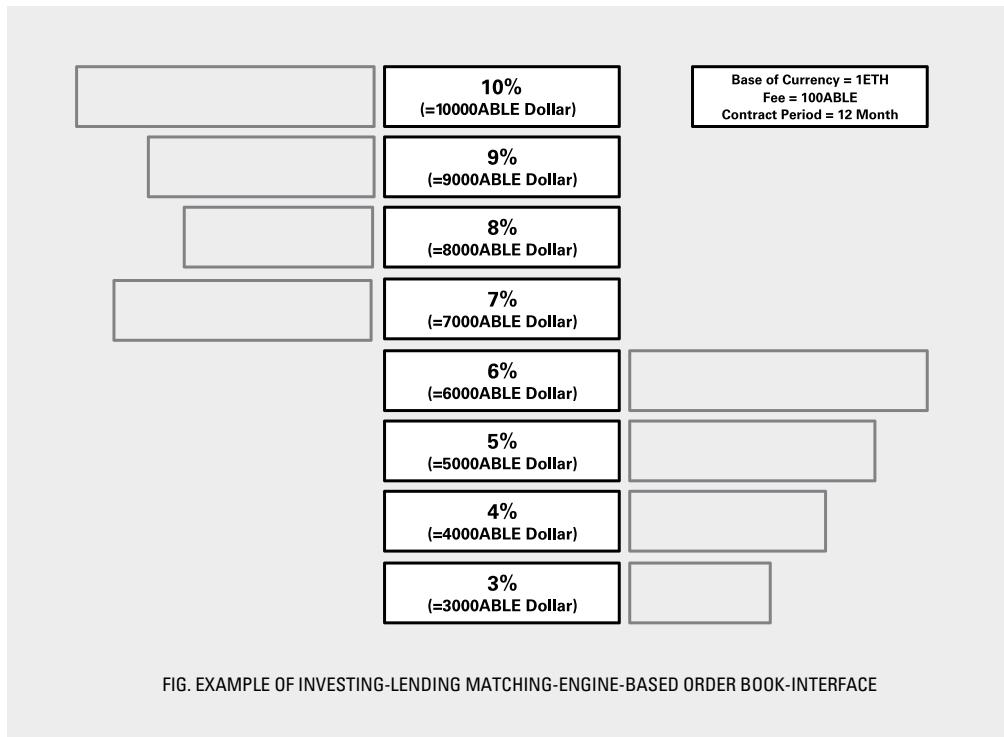
3.3 CONVENIENT CRYPTOCURRENCY ADDRESS AND SCHEDULED REMITTANCE SERVICE

3.4 ACCOUNT-BASED ICO SMART CONTRACT

03 ABLE PROJECT: FINANCIAL SERVICES

3.1 ABLE INVESTING/ LENDING MATCHING ENGINE

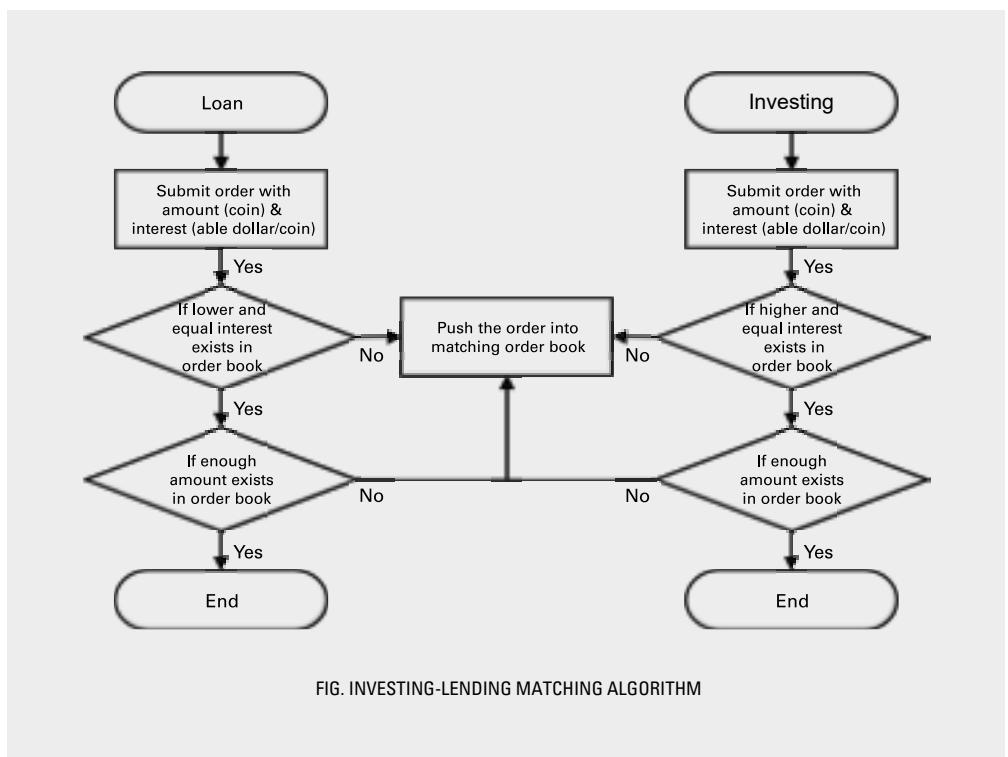
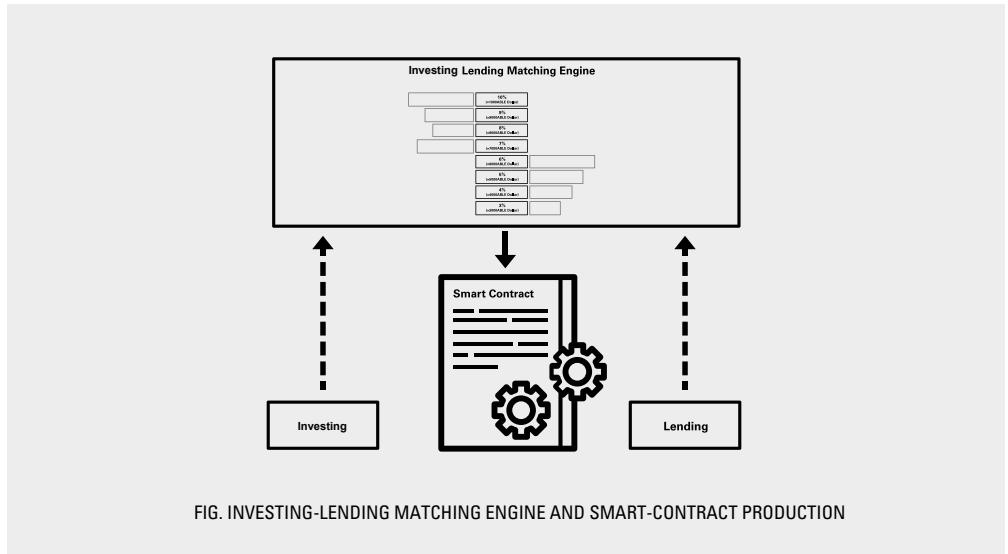
The ABLE project, in its initial stages, provides: 1) investing/lending – investing-lending matching engine; 2) payroll and credit score/ loan – cryptocurrency salary- payment service and credit scores to support loans; 3) simplified payment and remittance service – simplified account address service and reserve remittance; and 4) investment – account-based Initial Coin Offering (ICO) investment governed by a smart contract. The project offers an account-based finance experience that encompasses, loans, remittance, payroll, and investment.



The fundamental principle of decentralization is to share value among actual participants while eliminating unnecessary intermediaries in a given business model. The matching engine that directly links investors and borrowers on a peer-to-peer basis allows investors and borrowers to share actual value. Whereas in the traditional banking system, banks determine deposit and lending rates as intermediaries, the ABLE investing-lending matching engine allows a variety of investors and borrowers to participate in the process of determining appropriate rates.

Since the existing cryptocurrency banking market tended to focus on either deposits or loans, it was difficult to secure sufficient liquidity and trading volume. However, the market that enables direct linking on a peer-to-peer basis can increase liquidity and trading volume. Generating great amounts of various currency data made available to users, such as cryptocurrency short-term rates, long-term rates, and trading volume, reduces volatility and helps create a stable cryptocurrency market environment.

3.1.1 INVESTING-LENDING MATCHING ENGINE AND SMART CONTRACT



An order book for investing and lending is created based on investing and lending demand from users. Once an invest or lending transaction is carried out based on the order book, a smart contract is created to execute the investment or the loan. The borrower's cryptocurrency will be set up as collateral and the contracted loan will be given. Then the investor will receive corresponding interests.

3.1.2 INVESTING-LENDING MATCHING-ENGINE PRODUCTS

Investing-lending matching-engine products are created to increase liquidity and trading volume. Offering too many products prevents banks from securing sufficient liquidity and trading volume, and offering too few prevents them from meeting various demands. Considering that the cryptocurrency market cycle is shorter than the cycles of traditional financial markets, the solution will include products with relatively shorter cycles in its initial stages.

We plan to increase the number of cryptocurrencies supported (e.g., Bitcoin, Litecoin) just as the Atomic Swap feature is introduced.

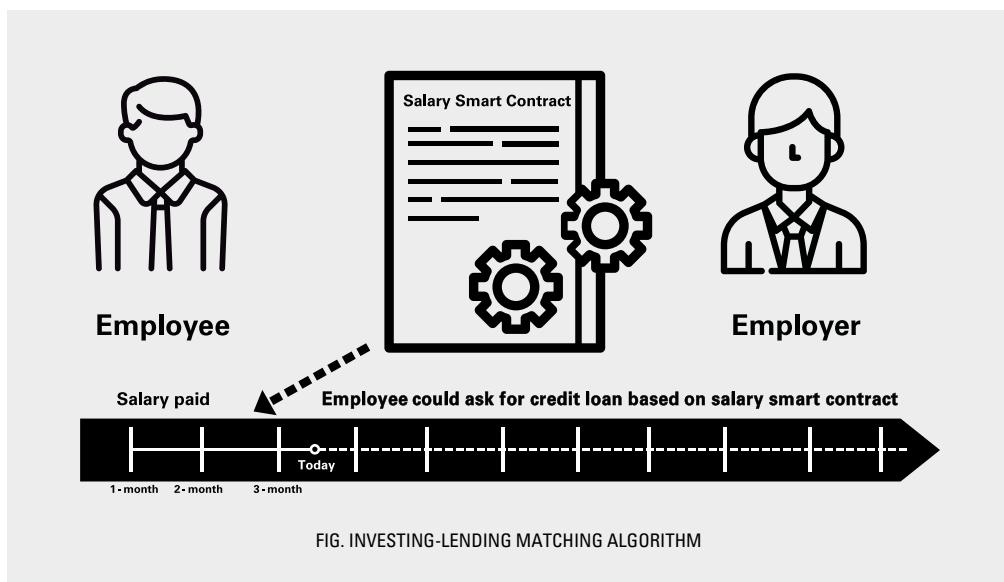
3.1.3 INVESTING-LENDING MATCHING-ENGINE BUSINESS MODEL

The entire investment process—investment money transfer, interest payment, and investment return at maturity—is carried out through smart contracts.

Collateralization of cryptocurrency, loan payment, and interest payment are also carried out through smart contracts. Users can invest any cryptocurrency set up as collateral in mutual funds listed on the ABLE network and various investment products, but cannot make withdrawals. Any investments made by cryptocurrency collateral can be automatically liquidated through smart contracts to preserve the value of collateral, in case value of the investments declines.

If the asset set up as collateral is PoS(Proof of Stake), users can use external master node services based on a smart contract.

3.2 CRYPTOCURRENCY PAYROLL AND CREDIT SCORE/LOAN



GMO, a Japanese Internet-based conglomerate, has paid some of its employee salaries in cryptocurrency. Growing interest in the cryptocurrency industry is expected to result in an increase in payments in cryptocurrency for some salaries or project costs.

Typically, loans are divided into secured loans backed by assets, and credit loans based on income. Currently, there are only few credit-loan services in the cryptocurrency space, because salary payment in cryptocurrency has received little consideration. However, as more salary payments are made in cryptocurrency, credit lending based on cryptocurrency also becomes possible. Salaries can be paid based on project progress or hours of work done, and credit assessment and credit lending can be serviced based on cumulative salary-payment data.

3.2.1 SALARY PAYMENT SMART CONTRACT

Personal labor can be assessed largely based on hours of work and performance. Initially, the salary payment system will focus on assessing hours of work. Then salaries can be paid on a yearly, monthly, or weekly basis.

3.2.2 CREDIT ASSESSMENT

Credit assessment can rest on the fact that blockchain data cannot be manipulated. In initial stages, credit scores are generated based on the history of completion of salary-payment smart contracts and consistency of cryptocurrency income. Then, based on the credit scores, the option to allow salary advance payments will be added. Credit scores will be refined by credit-assessment technology based on big data and external credit-score data.

3.2.3 CREDIT LOAN BASED ON SMART CONTRACT FOR SALARY PAYMENT

Adding the option of making a credit loan to the cryptocurrency salary payment system allows users to take out a credit loan based on existing smart contracts for salary payment, using the credit-score data within the account.

3.3 CONVENIENT CRYPTO-CURRENCY ADDRESS AND SCHEDULED REMITTANCE SERVICE

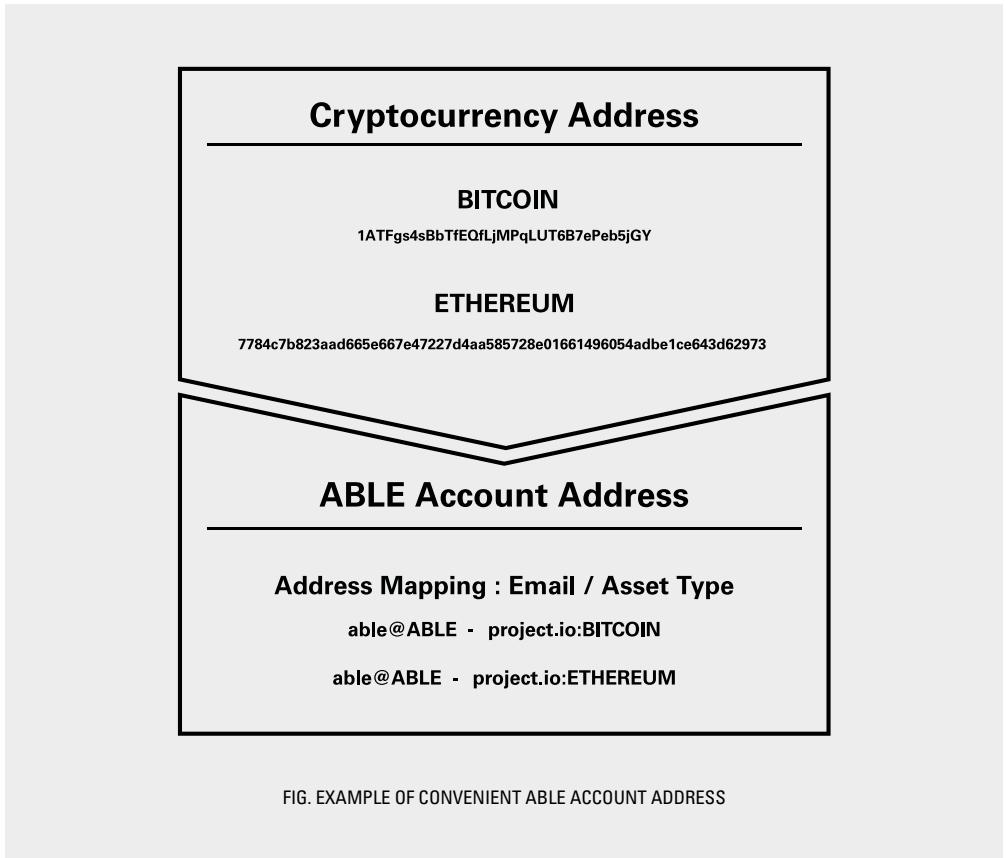


FIG. EXAMPLE OF CONVENIENT ABLE ACCOUNT ADDRESS

The address lengths of cryptocurrencies are typically 34 characters for Bitcoin and 65 characters for Ethereum, to prevent hash collisions and hacking. However, a remittance service that requires long addresses is a hassle for users. Making cryptocurrency remittance more accessible means making cryptocurrency addresses convenient, a principle similar to that by which web-based applications use easily remembered domain URLs instead of IP addresses.

We provide convenient addresses within the ABLE account to boost remittance activities, including small-amount payments. Additionally, we provide the option that allows users to cancel remittance at any time before the reserved time through the scheduled remittance service.

3.4 ACCOUNT-BASED ICO SMART CONTRACT

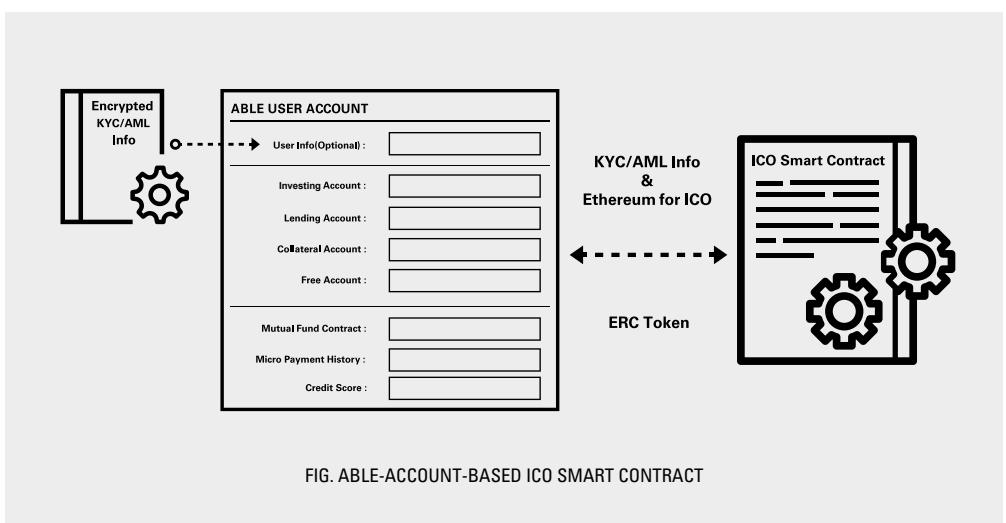


FIG. ABLE-ACCOUNT-BASED ICO SMART CONTRACT

Currently, solutions must provide a variety of data for KYC/AML(Know-Your-Customer/Anti Money Laundering) to be effective in ICO. Files and personal data can be stored in the ABLE account through encryption so that users can invest in ICO on the ABLE account using personal data and an ICO smart contract.

04 ABLE ECOSYSTEM DEVELOPMENT PLAN

4.1 ABLE DEX

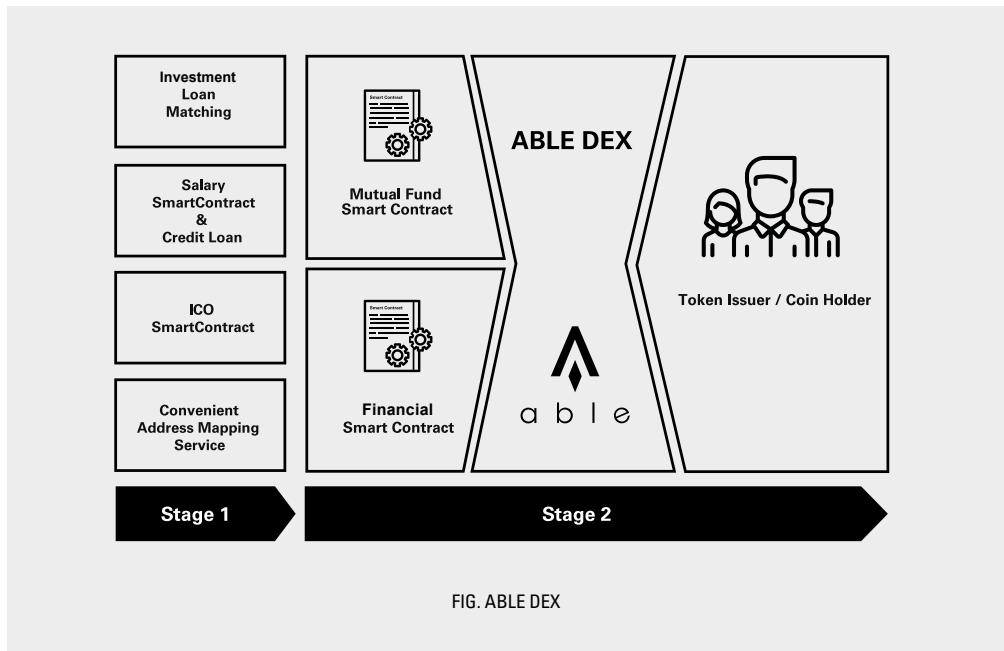
4.2 ABLE FINANCIAL PRODUCTS

4.3 MICROPAYMENT / THIRD PARTY

04 ABLE ECOSYSTEM DEVELOP- MENT PLAN

The ABLE Ecosystem largely comprises the ABLE account and support for third parties providing finance, wealth management, and micropayment services. We will add the decentralized exchange (DEX) feature to store smart contracts based on various cryptocurrencies. Operating on DEX, we will distribute the framework to allow production of wealth management and finance smart contracts, creating an environment where a wide array of product developers can develop financial products. Thus, users can access a variety of financial products, and through API development the ABLE project can evolve into a cryptocurrency finance platform and an ecosystem enabling communication, including micropayment, with outside parties.

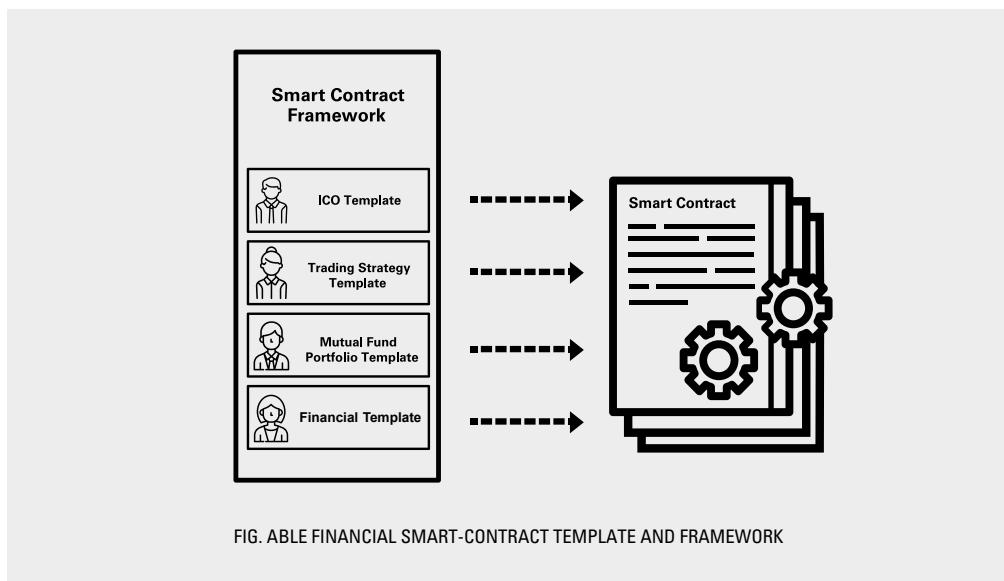
4.1 ABLE DEX



We need a decentralized exchange through which cryptocurrency assets can be traded in order to develop wealth management and finance smart contracts. Currently, most cryptocurrency investments are made through arbitrage trading at centralized exchanges or ICO participation with personal wallets. However, through ABLE DEX, we can eliminate the risks of centralized exchanges and allow for wealth management.

Through the ABLE account, we can create a platform that makes a variety of investment products accessible. On the ABLE network, we will create an environment where various smart contracts and Decentralized Applications (DApps) can enable the development of the ABLE Ecosystem.

4.2 ABLE FINANCIAL PRODUCTS



Utilizing various smart-contract frameworks and templates, users can develop cryptocurrency investment products. Using these templates, financial-product developers can create a variety of wealth management and finance smart contracts without having to learn difficult programming languages. Users can access various investment products through the ABLE account.

4.2.1 SMART CONTRACT FOR WEALTH MANAGEMENT

We develop the smart contract frameworks necessary to manage funds, such as trading strategy generators, and portfolio analysis tools. Fund-management service fees will attract investment-product developers and create an ecosystem that will:

- 1) Allow creation of a variety of technical indicator-based trading strategies.
- 2) Create a systematic wealth management environment by providing smart contracts that enable portfolio construction and performance analysis tools.

4.2.2 FINANCE SMART CONTRACT

While the investing-lending matching engine carries out most loans, demand for various financial services will be met through finance smart contracts that include long-term loans, large-amount loans, P2P loans, secured loans, and credit loans.

Secured loans, and credit loans based on external ABLE data will be carried out through various integration mechanisms, following the development of the ABLE main network.

Initially, ABLE will produce finance smart contracts; then ABLE will gradually provide an environment in which various financial-product developers can produce finance smart contracts based on the finance smart contract framework.

4.3 MICROPAYMENT/ THIRD PARTY

A big portion of financial activities such as credit-card payments, automatic transfers of utility and service charges, and mileage programs will be integrated into the cryptocurrency space.

Extending the scope of practical application based on a variety of partnerships with third-party services, including micropayment, will strengthen the ABLE ecosystem. In the future, a majority of economic activities will be integrated into the cryptocurrency space by cryptocurrency finance.

05 ABLE SYSTEM ARCHITECTURE

5.1 ABLE SYSTEM

5.2 ABLE ACCOUNT

5.3 ABLE CRYPTOCURRENCY ECONOMIC SYSTEM

5.4 ABLE CONSENSUS PROTOCOL

05 ABLE SYSTEM ARCHITEC- TURE

5.1 ABLE System

Users can use the ABLE account to access third-party services including micropayment, investments, loans, and wealth management. This strategy allows existing bank-account users engaged in financial activities to expand their services in more accessible ways. We aim to provide users with financial services integrated around accounts.

5.2 ABLE Account

ABLE USER	
User Info(Optional) :	<input type="text"/>
Investing Account :	<input type="text"/>
Lending Account :	<input type="text"/>
Collateral Account :	<input type="text"/>
Free Account :	<input type="text"/>
Mutual Fund Contract :	<input type="text"/>
Mutual Payment History :	<input type="text"/>
Credit Score:	<input type="text"/>

ABLE ACCOUNT	
Account info(period, desc) :	<input type="text"/>
Account Owner :	<input type="text"/>
Account Number :	<input type="text"/>
Password :	<input type="text"/>
Account Email :	<input type="text"/>
Account Type :	<input type="text"/>
Coin Type :	<input type="text"/>
Balance :	<input type="text"/>

FIG. ABLE USER STRUCTURE

FIG. ABLE ACCOUNT STRUCTURE

All the services of the ABLE system will operate on smart contracts based ABLE User and ABLE Account. The ABLE User can have one or more ABLE accounts, depending on the purpose. The ABLE User data includes user information, a list of accounts held, fund/loan subscription details, payment history, and creditworthiness. The ABLE Account encompasses deposit account, loan account, collateral account, free account, fund/finance account, and credit score. Each ABLE Account includes account information, account holder, account number, password, e-mail address linked to the account, account type, deposit coin, and balance.

Upon creation of the ABLE account, the user can transfer and withdraw money through the ABLE free account, and utilize various products based on smart contracts.

5.3 ABLE CRYPTOCURRENCY ECONOMIC SYSTEM

5.3.1 ABLE COIN/ABLE DOLLAR

ABLE currency consists of ABLE Coin and ABLE Dollar. ABLE Coin is used as a service charge for using the ABLE system, and ABLE Dollar is used for interest payment. ABLE Coin and ABLE Dollar can be exchanged.

5.3.2 ISSUING ABLE COIN/ABLE DOLLAR

The amount of ABLE coins initially issued is 25 billion, and the token decimal unit is 18. Validation nodes will be operated on a PoS basis. The additional inflation rate through initial PoS-based operation is 15%, which will converge into 5% in the long run. Of the outstanding tokens, 15 billion will be distributed to the general public and 10 billion will be assigned to related parties.

The initial amount of ABLE Dollar issued is 1 billion. To ensure minimum volatility, no additional ABLE Dollar will be issued.

ABLE adopts the PoS consensus protocol. The minimum number of ABLE coins preserved to operate validation nodes will be 20 million.

5.4 ABLE CONSENSUS PROTOCOL

06 TECHNICAL STRUCTURE OF ABLE SYSTEM

- 6.1. ARCHITECTURE OF ABLE SYSTEM
- 6.2. APIs OF ABLE SYSTEM
- 6.3. ABLE COIN ON ABLE SYSTEM

06 TECHNICAL STRUCTURE OF ABLE SYSTEM

The development of ABLE system goes through three stages. At the first stage, we first develop ABLE account, investing-lending matching, and convenient cryptocurrency address services through solidity-based ABLE smart contract technology on the Ethereum blockchain and then provide ABLE service through the ABLE web page. At the second stage, we develop ABLE DEX and design various ABLE financial products that interact with ABLE smart contracts and provide them to users through the web page. At the final stage, we create a dedicated blockchain main network for ABLE that allows us to provide credit services and personalized financial products by gathering and analyzing user patterns of using financial products.

6.1. ARCHITECTURE OF ABLE SYSTEM

The ABLE system has the following architecture: it is made up of infrastructure layers, ABLE framework layers, and application layers, and members have limited access to the respective layers depending on their access level.

INFRASTRUCTURE

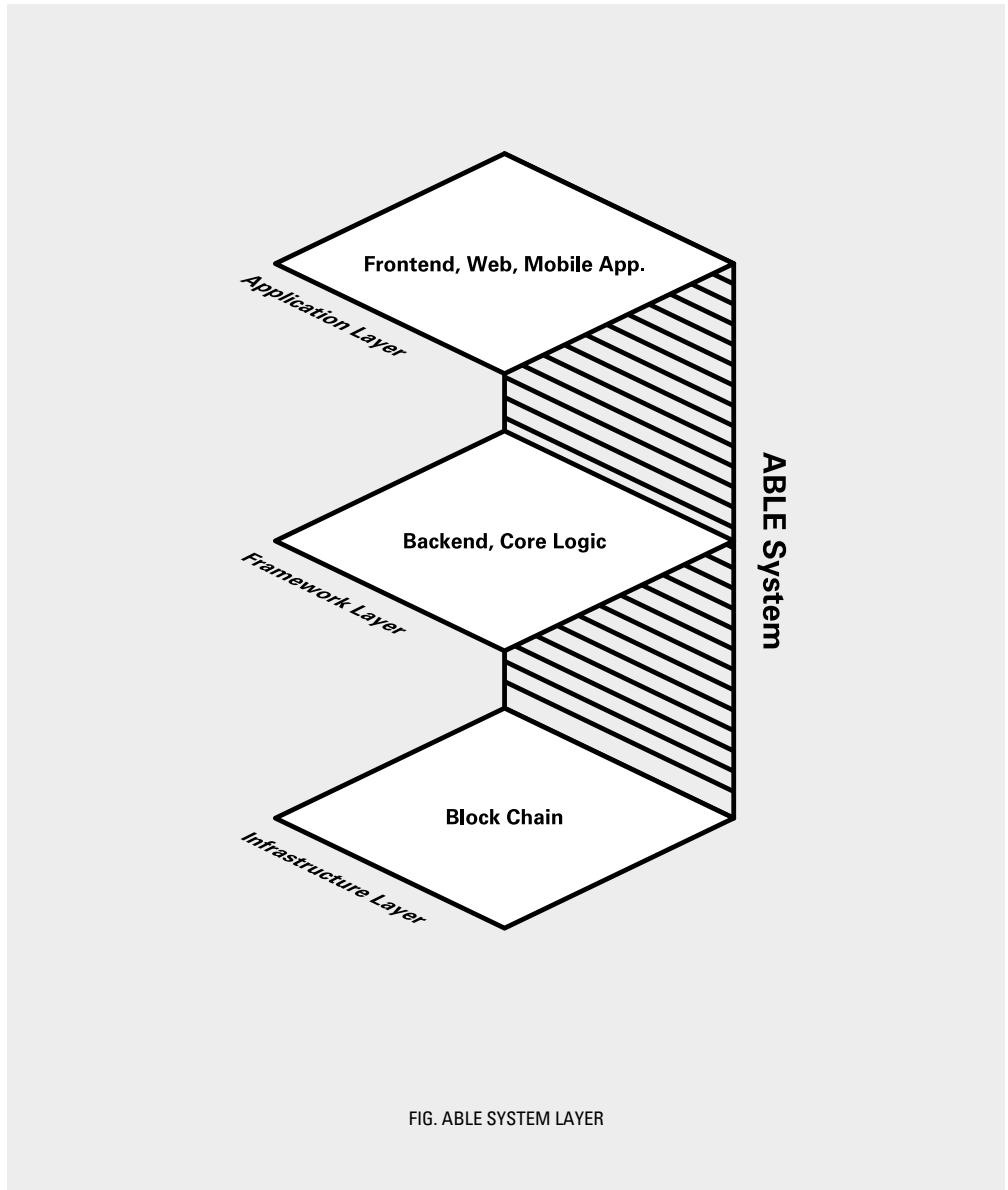
The layer consists of blockchain systems for managing shared log and executing smart contracts. In the initial stages of the ABLE project, Ethereum blockchain will be used and then replaced by dedicated ABLE blockchain. In the infrastructure layer, ABLE accounts, investing-lending matching engines, the DEX, and cryptocurrency gateways are designed, forming the foundation of the ABLE system. Access to the infrastructure layer is limited to the ABLE system, but the layer is maintained and repaired at the request of users.

ABLE FRAMEWORK

The framework provides cryptocurrency gateways and the DEX to allow for exchange of cryptocurrencies between different blockchain systems and offers finance solutions for investing-lending matching engines and ABLE products. Smart contracts of ABLE financial products are designed based on their characteristics. Access to the framework layer is limited to the ABLE system which validates and distributes such financial products.

ABLE APPLICATION

The application introduces how to transfer money, create an account and use products designed based on smart contracts and delivers user requests to the framework. Using account smart contracts, users can create their own ABLE account, transfer money, and use financial products through distributed product smart contracts. The application layer provides web sites and smartphone apps for users to easily use the ABLE system. Access to the application layer is also given to users, and they can access the ABLE finance and financial products through the web and applications.



The figure below represents the structure of the ABLE system, which will be realized on ABLE blockchain, and the API relationships among the layers. Users can use the ABLE system through the website and mobile applications in the ABLE application layer. All the actions requested by users are delivered to the ABLE Framework through public APIs, and smart contracts are called at such requests. All requests among smart contracts within the ABLE framework layer are carried out through protected API. The ABLE system builds finance and account smart contracts which are responsible for ABLE account-related processing, transfer, and remittance transactions. Then smart contracts are designed to provide services for investments, loans, and financial product transactions. The ABLE infrastructure layer manages all smart contract codes working on the ABLE framework and users' account transaction details, remittance details, and financial product details in blockchain. The ABLE blockchain learns users' service details through big data analysis and recommends financial products tailored to users' patterns.

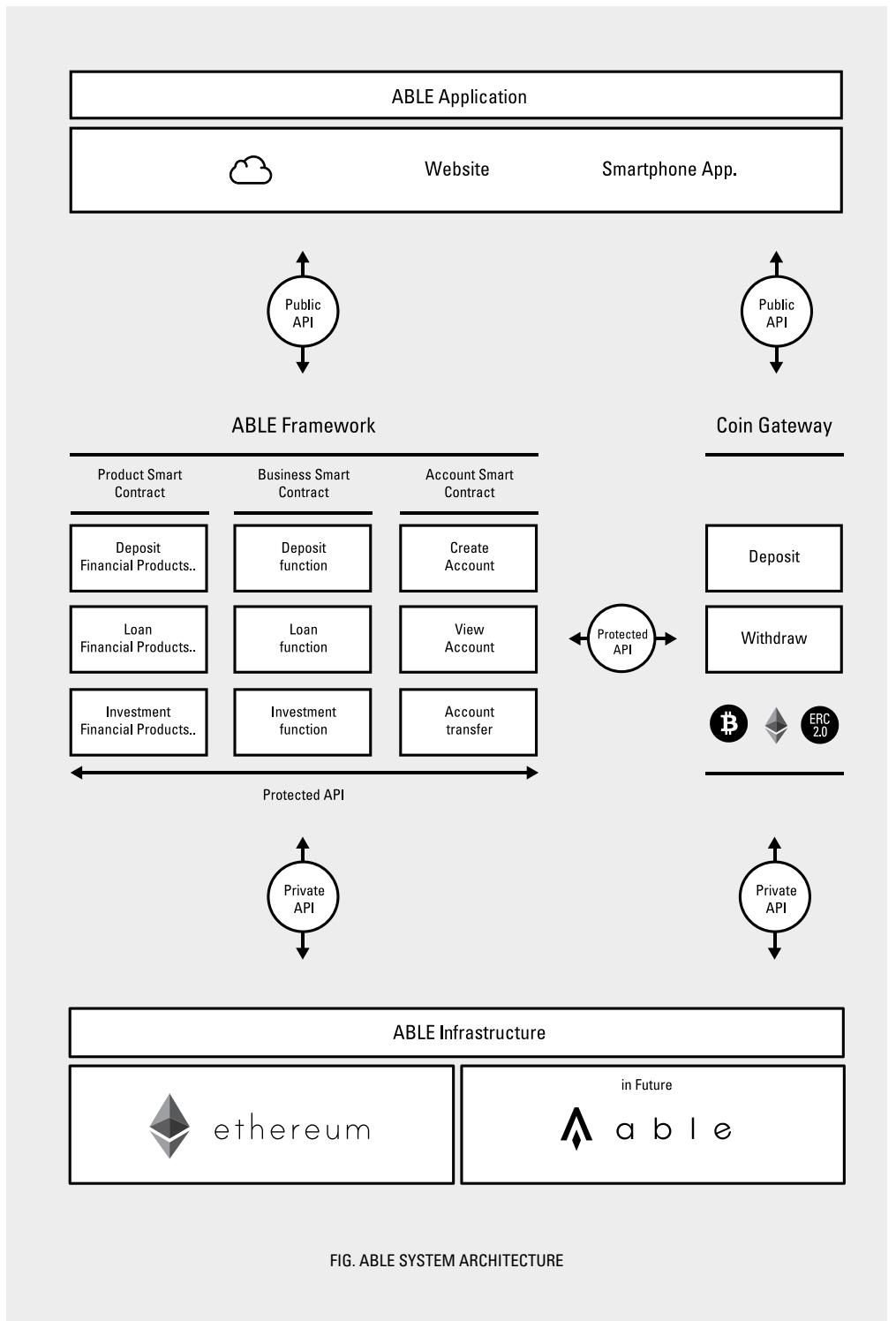


FIG. ABLE SYSTEM ARCHITECTURE

6.2. APIs OF ABLE SYSTEM

Since the ABLE system architecture is layered, and levels of access vary depending on the layer, all respective layers and components communicate through standard-based ABLE APIs. The respective APIs are divided into public API, protected API, and private API, and levels of access to respective APIs are divided into users and the ABLE system.

PUBLIC API

REQUEST_DEPOSIT(COIN TYPE, AMOUNT)

Used when users deposit cryptocurrency money. Deposits are made based on the type and the amount of coins deposited.

REQUEST_WITHDRAW(COIN TYPE, DESTINATION ADDRESS, AMOUNT)

Used when users withdraw cryptocurrency money. Withdrawals are made based on the type of cryptocurrency withdrawn, the address of cryptocurrency to receive, and the amount of coins.

REQUEST_TRANSFER(SOURCE COIN TYPE, DESTINATION COIN TYPE, DESTINATION ADDRESS, AMOUNT)

Used when users transfer cryptocurrency money between accounts. Transfers are made based on the type of cryptocurrency transferred, the address to transfer cryptocurrency to, and the amount of coins. Exchange between different cryptocurrencies will be made at the exchange rates and service charges set by the ABLE system.

REQUEST_SMARTCONTRACT(COIN TYPE, PRODUCT ADDRESS, AMOUNT, PARAMETERS)

Used when users apply for financial products. Request for financial products is made based on the type of cryptocurrency sent, the address of the smart contract for chosen products, and quantities.

PROTECTED API

START_SMARTCONTRACT (COIN TYPE, SOURCE ADDRESS, SMARTCONTRACT ADDRESS, AMOUNT, PARAMETERS)

When users request financial products, it sends the type and the amount of coins requested for the smart contract in question. When the smart contract expires or Stop_Fund occurs, End_Fund will be executed.

STOP_SMARTCONTRACT (COIN TYPE, SOURCE ADDRESS, SMARTCONTRACT ADDRESS)

When users cancel financial products, the coins put into the product will be returned through smart contracts. Interim payments will be made under the agreement for the product.

END_SMARTCONTRACT (COIN TYPE, DESTINATION ADDRESS, AMOUNT)

When smart contracts expire or are terminated, the result will be sent based on the type of products to the account of users who requested the products.

EXCHANGE_COIN(SOURCE COIN TYPE, DESTINATION COIN TYPE, AMOUNT, SOURCE ADDRESS, DESTINATION ADDRESS)

Used for transfer of cryptocurrency between accounts or used to exchange cryptocurrency when smart contracts expire.

SET_SMARTCONTRACT(SMARTCONTRACT NAME, SMARTCONTRACT DESCRIPTION, SMARTCONTRACT ADDRESS)

Used when registering financial product smart contracts designed with the ABLE system. The product thus registered is made accessible to users through the application layer.

DELETE_SMARTCONTRACT(SMARTCONTRACT NAME)

Used to delete any expired financial product smart contracts.

PRIVATE API

ETHEREUM APIs

These APIs are fundamental to the Ethereum blockchain, and the ABLE Framework is created using Ethereum APIs. Ethereum APIs are called by requests.

ABLE APIs

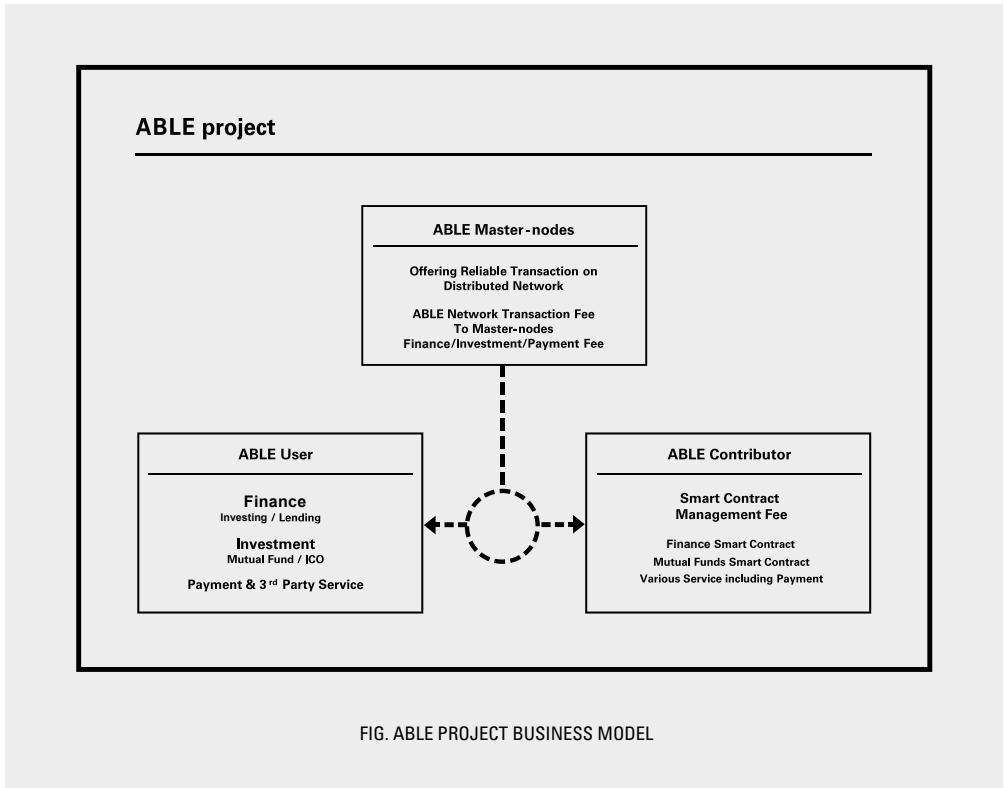
These APIs are dedicated ABLE blockchain APIs and basically include the same functions as Ethereum APIs. They include additional APIs that can process personal data and details of customers that are difficult to register with the public Ethereum blockchain.

6.3. ABLE COIN ON ABLE SYSTEM

ABLE coins are designed to maintain the ABLE system and used by vendors. When a new financial product smart contract is registered with the ABLE framework, ABLE coins are used as a Gas fee, which is a registration fee. Thus to release new financial products, ABLE coins should be continuously purchased from users and vendors, and the continuous purchase of ABLE coins to maintain the system prevents dumping of ABLE coins. Next, when users exchange cryptocurrencies through the ABLE DEX, ABLE coins are used as service charges. Lastly, users can use ABLE coins to pay for products or services offered by ABLE vendors.

07 BUSINESS MODEL

07 BUSINESS MODEL



The ABLE project is largely composed of ABLE User, Contributor, and Master-nodes. The user can use financial services, investment products, payments, and a variety of services. The contributor produces a wide array of services including finance, investment products, and payment that the user can utilize, and receives service charges. The master-nodes ensure stability in operating a decentralized ABLE network based on PoS consensus algorithms, and receive network-transaction service charges.

08 CONCLUSION

08 CONCLUSION

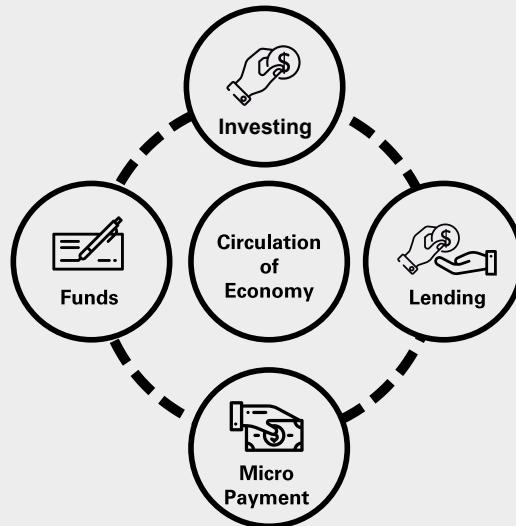


FIG. ABLE ECOSYSTEM SCHEME

The ultimate goal of ABLE is to provide users with a great user experience, while simultaneously allocating value to ABLE through all the benefits described above. In finance, the project aims to create a systematic ecosystem and develop a better user-accessible system by applying blockchain technology to any area deemed necessary. ABLE intends to integrate into one system all the financial solutions currently dispersed across the blockchain market, by offering a solution to existing banks' problems, and an alternative to centralized deposit-lending systems.

The P2P investing-lending system, which is the core of the ABLE project, is particularly significant because it creates personalized accounts for cryptocurrency assets. Existing micropayment and fund solutions suffer from numerous shortcomings due to the lack of an integrated account, and users have substantial trouble using those solutions. Additional solutions will be created in the integrated system to provide various services to users through their account designed for them.

The ABLE project aims to improve accessibility for the general public to easily use institutional-grade products with low service charges and high returns on investments, as well as low lending rates, by utilizing the benefits of blockchain technology. Personal accounts will enable users to receive transparent services difficult to earn through personal wallets. Through the account and micropayment system, cryptocurrencies can be used to make actual payments, and users can participate in smart contract-based financial products. Financial products provided by ABLE offer increased access to products in which individuals have difficulty making investments. ABLE plans to allow users to use products to which access is typically limited, such as the presale of unlisted coins with relatively high minimum-investment amounts; and intends to provide users with the opportunity to actually utilize cryptocurrencies.

By applying blockchain technology to various financial services, such as depositing cryptocurrency, using cryptocurrency coins in real life, and making investments with cryptocurrency, we will create a system that meets the expectations of those anticipating the fourth industrial revolution in this age of blockchain.

NOTICE & CAUTION

The ABLE project Whitepaper has been prepared to outline the project and provide a specific description of the roadmap. This Whitepaper has not been prepared to encourage or attract investment, and all readers of this Whitepaper should be advised that ABLE token issuer and service provider, Chain Holdings OÜ (registration code 14406869) and its Korean agency, K-Blockchain (registration code 462-86-00783) are not responsible for any loss, damage, or obligation, financial or otherwise, arising from the reader consulting with this Whitepaper. Please be advised once again that Chain Holdings OÜ and K-Blockchain accept no responsibility or liability for any financial damage, loss, or obligation arising from the reader's use of this Whitepaper in any decision-making process, including, without limitation, consulting with this Whitepaper and basing decisions on this Whitepaper. Values or stabilities of any ABLE tokens including ABLE Coin and ABLE Dollar are not guaranteed.

This Whitepaper for ABLE project is provided on an as-is basis and does not warrant that any information contained herein will remain accurate or unchanged until a certain future point in time.

Chain Holdings OÜ and K-Blockchain make no representations or warranties to any reader of this Whitepaper and disclaims all legal liability related to this Whitepaper. For example, Chain Holdings OÜ and K-Blockchain make no warranties that this Whitepaper has been prepared based on legal rights; that it constitutes infringement of third-party rights; that it has any commercial value or use; that it serves specific purposes that readers of this Whitepaper have in mind; or that it contains no errors. The extent of this disclaimer is not limited to the examples illustrated above.

REFERENCES

Lending-deposit spread business at world banks,
<https://fred.stlouisfed.org/series/DDEI021WA156NWDB>

SALT Whitepaper,
<https://membership.saltlending.com/files/abstract.pdf>

British Government Ponzi Fraud Cryptocurrency Service Sanctions,
<https://beta.companieshouse.gov.uk/company/10278342/filing-history>

Atomic Swap,
https://en.bitcoin.it/wiki/Atomic_cross-chain_trading

Japan's GMO Internet Group Will Pay Thousands of Workers in Bitcoin,
<https://bitcoinmagazine.com/articles/japans-gmo-internet-group-will-pay-thousands-workers-bitcoin/>

Bitcoin 34-character address length,
<https://blockexplorer.com/>

Ethereum 65-character address length,
<https://etherscan.io/>

IPv6,
https://en.wikipedia.org/wiki/IPv6_address

KYC (Know Your Customer),
https://en.wikipedia.org/wiki/Know_your_customer,
Money Laundering,
https://en.wikipedia.org/wiki/Money_laundering



a b l e

ABCC

A Digital Asset Exchange



www.abcc.com

Index

Our Vision	2
1. Background	3
1.1 Opportunities	3
1.2 Pain Points to be Addressed	3
1.2.1 A Lack of Prudent Digital Asset Evaluation Frameworks	3
1.2.2 A Shortage of Variety for Digital Assets	4
1.2.3 Security Issues	4
2. Our Business	6
2.1 Business Model	6
2.1.1 Digital Asset Evaluation Framework	6
2.1.2 Digital Asset Innovation	6
2.1.3 Early-stage Blockchain Project Investment and Incubation	7
2.1.4 Features	7
2.1.5 Technical Architecture and Trading Environment	9
2.1.6 Capital Safety and Risk Management	10
2.2 Revenue Model	11
2.3 Roadmap	11
3. About Us	12
3.1 Our Values	12
3.2 Our Competitive Advantages	12
3.2.1 Industry Leading Technologies	12
3.2.2 Finance Veterans from Top Financial Institutions	12
3.2.3 Global Vision and Experience	13

Our Vision

Satoshi Nakamoto, the pseudonymous "creator" of Bitcoin, published his Bitcoin White Paper "Bitcoin: A Peer-to-Peer Electronic Cash System" in 2008 and subsequently mined the Genesis Block in 2009. Vitalik Buterin launched Ethereum, which can execute smart contracts, in 2015. Blockchain technology has been disrupting a wide range of industries at an exponential speed.

Decentralized and distributed, blockchain technology is offering us fundamentally different solutions to many problems. It helps build trust and consensus based on cryptography rather than relying on a trusted central party so that economic activities can take place and capital can flow across borders frictionlessly. As blockchain-based asset digitalization is becoming a new trend, more and more investors are adding digital assets to their investment portfolios. There is increasing consensus to democratize the digital economy so that everyone can participate and enjoy the benefits brought about by blockchain technology.

ABCC aims to build and nurture a blockchain-enabled digital asset platform. We endeavor to build bridges between digital asset users, developers and investors to facilitate effective information flow and value creation. We seek to help visionary digital asset investors achieve their investment objectives and assist promising blockchain project teams secure funding and gain market recognition by providing select digital assets, designing and operating innovative digital assets, and investing in and supporting blockchain projects with high growth potential. Ultimately, our goal is to promote a healthy and sustainable growth of the blockchain industry.

1. Background

1.1 Opportunities

Since its inception, the internet has facilitated information flow and sharing by establishing a wide and efficient network. Built on top of the internet and as an upgrade, blockchain technology has established a decentralized trust as opposed to a centralized one. It has enabled tamper-resistant record keeping and distributed information storage and sharing by integrating advanced cryptography and database technologies.

As a unique financing solution and a native part of most blockchain projects, cryptocurrency has experienced rapid growth both in terms of its overall market capitalization and its variety. Subsequently, mainstream investors, both individual and institutional, are increasingly including cryptocurrencies as part of their investment portfolios.

Serving as a key part of the digital asset ecosystem, digital asset exchanges connect different stakeholders, e.g., blockchain project teams, investors, and advisors, to optimize the resource allocation in this growing industry.

We aim to contribute to the healthy and sustainable growth of the blockchain ecosystem. Optimistic about the potentials of digital asset exchanges, we endeavor to build and nurture long-term trusting relationships with visionary investors, project teams, and other stakeholders in the blockchain ecosystem.

1.2 Pain Points to be Addressed

1.2.1 A Lack of Prudent Digital Asset Evaluation Frameworks

Albeit fast growing, the blockchain industry is still in its infancy with numerous emerging projects and cryptocurrencies. While some of these projects are truly disruptive, some are merely concepts or even scams.

As it is an emerging and burgeoning industry, there is a lack of generally accepted evaluation frameworks for digital assets. With a relatively limited understanding of

blockchain technology and cryptocurrencies, retail investors have been exposed to high risks.

While there are more and more digital asset exchanges, not many of them have adopted a clear and systematic approach towards digital asset evaluation. With sometimes lucrative listing fees, some digital asset listing decisions have been made from a commercial point of view rather than based on the quality of the assets itself. Hence, it is increasingly pressing and challenging to ensure that only select high-quality digital assets are presented and provided to investors.

1.2.2 A Shortage of Variety for Digital Assets

Compared with traditional financial instruments, there is a shortage of variety for blockchain native digital assets. More sophisticated investment vehicles such as margin trading, investment portfolios, futures, options and other asset-backed securities based on cryptocurrencies, are still nascent.

As the digital asset industry grows and gradually matures, some of the world's leading digital asset exchanges are rolling out their crypto derivative products on top of their mainstream crypto trading offerings. However, financial derivatives are complex in design and risky in nature. Only finance professionals with a rich experience in derivatives are capable of successfully designing and operating such products. Meanwhile, systems and procedures have to be deployed to ensure the safety and security of digital asset trading platforms so that investors' interests are safeguarded.

1.2.3 Security Issues

Along with a lack of prudent digital asset evaluation frameworks and a shortage of variety for digital assets, security is another issue of top concern for investors.

On the one hand, security is still a challenging issue for blockchains on the infrastructure layer. Cryptocurrencies may suffer from different attacks such as "51% Attack", threatening the integrity of and the foundation upon which the trust and consensus of blockchain technology are built. For example, Verge, a cryptocurrency for users who value security and privacy, suffered from a "51% attack" due to a bug in

its codes. As a consequence, its price dropped by 15% within 24 hours after the attack.

On the other hand, cybersecurity incidents at digital asset exchanges happen every now and then. Mt. Gox, a cryptocurrency exchange based in Japan and the then biggest one in the world, was hacked in March 2014 and as a result, Bitcoin worth some \$473 million at the time—and representing 7% of all Bitcoins then in existence—had disappeared. On June 22nd, 2018, the exchange was finally allowed to swap bankruptcy proceedings to civil rehabilitation. Similarly, digital assets equaling \$31 million were hacked from Bithumb, a cryptocurrency exchange based in South Korea.

Cybersecurity practices at a digital asset exchange are based on both of its technological and operational capabilities and are demanding for both. According to "2017 Global Cryptocurrency Benchmarking Study" by Cambridge Judge Business School, there are a few key findings on exchanges' security practices.

- On average, security headcount corresponds to 13% of total employees; 17% of the budget is spent on security;
- 80% of large exchanges and 69% of small exchanges use external security providers;
- Optional two-factor authentication (2FA) is offered to customers by a majority of exchanges and required for employees for most operations;
- 79% of exchanges provide regular security training programs to their staff;
- 92% of exchanges use cold-storage systems; on average 87% of funds are kept in cold storage;
- Multi-signature architecture is supported by 86% of large exchanges and 76% of small exchanges;
- 60% of large exchanges have external parties performing their formal security audits, while 65% of small exchanges perform them internally.

2. Our Business

With blockchain technology disrupting a wide range of industries, ABCC is well positioned to solve the pain points and challenges presented and elaborated above. We endeavor to provide a secure and convenient digital asset exchange platform with our proven technological and operational capabilities.

2.1 Business Model

2.1.1 Digital Asset Evaluation Framework

ABCC aims to build and nurture a blockchain-based digital asset exchange. We endeavor to build bridges between digital asset users, developers and investors to enable effective information flow and value creation. We will help visionary digital asset investors obtain risk-adjusted returns and grow their assets. At the core of this ecosystem is our proprietary digital asset evaluation framework.

We assess prospective digital assets with our proprietary multi-dimensional evaluation framework. Before reaching an asset listing decision, we will comprehensively assess factors such as product/services, industry attractiveness, competition, regulations, team, technology and operations, crypto economics, liquidity, community, and others.

2.1.2 Digital Asset Innovation

By leveraging our deep and broad experience in financial innovation, we will gradually roll out our innovative digital asset offerings to enable our users to optimize their digital asset allocation and manage their risk exposures. Once we have obtained relevant licenses and permissions, we plan to provide various innovative digital asset products such as margin trading and contract trading.

Users can enter an order to borrow a desirable amount of funding from us so that they can raise their trading positions with leverage. Likewise, users will have the option of borrowing digital assets from us and selling them when they deem the

price to be high. They can buy back the assets and repay us when the price reverts to a healthy level.

We will help our users to achieve their investment objectives flexibly by offering digital asset futures and options. On one hand, this will offer hedging tools to our users to manage the risk exposures of their current digital asset positions. On the other hand, this will enable our users to exploit investment opportunities presented in the mispricing of certain digital assets based on their judgments. Ultimately, we will help investors achieve their optimal risk-adjusted investment returns.

Further, by investing in investment portfolios and other asset-backed securities based on cryptocurrencies, users can achieve their optimal risk-adjusted returns with the help of professional digital asset analysis and portfolio construction.

At the same time, we have rigorous systems and procedures to ensure the safety and security of our digital asset trading platform so that investors' interests are safeguarded.

2.1.3 Early-stage Blockchain Project Investment and Incubation

We plan to take an active role in early-stage blockchain projects by making strategic investments and providing incubation and enabling services. This will help projects with high growth potential to secure funding and gain market recognition while providing premium investment opportunities to our users.

2.1.4 Features

We aim to provide a wide range of blockchain native digital assets and other related services.

Crypto Trading

One of our core product offerings is crypto trading. We select our listed digital assets with our proprietary multi-dimensional digital asset assessment framework and ensure that they are presented in a visually friendly way. In this way, our users can achieve their investment objectives efficiently.

Fiat Trading

Once we have obtained relevant licenses and permissions, we plan to roll out fiat trading services to our users. This will further improve and optimize our secure and seamless trading services.

Margin Trading

Once we have obtained relevant licenses and permissions, we plan to provide margin trading services to our users. With leverage, our users will be able to achieve more efficient and flexible asset allocation.

Contract Trading

Once we have obtained relevant licenses and permissions, we plan to roll out contract trading services to our users, e.g., futures, options, investment portfolios and other asset-backed securities based on cryptocurrencies. This will enable digital asset investors to manage their risk exposure and obtain risk-adjusted returns.

OTC (Over the Counter)

We actively trade on mainstream digital asset exchanges and provide block trading services over the counter. Our trading partners include Ultra High Net Worth Individuals, family offices, mining businesses, hedge funds and other traditional financial institutions.

Value-added Services

We will leverage our knowledge and expertise in blockchain technology and digital assets to provide value-added services to facilitate our users' informed investment decision-making. Our value-added services will include but are not limited to news push, asset allocation services, and other self-services.

Devices

We will offer our digital asset trading services across multiple devices and platforms, including PC Web, Mobile HTML5, iOS, Android, and others.

2.1.5 Technical Architecture and Trading Environment

High-Performance Matching Engine

Our exchange platform has adopted a pure in-memory matching technology, with which a single machine can facilitate millions of simultaneous matchings. With cluster-sharding, this number can be further increased to over 10 million so that efficient low-latency matching can be a reality.

Meanwhile, our technical architecture is highly stable with hot failover based on our experience and proven track record at BAT, i.e., Baidu, Alibaba and Tencent.

Extendable Multiple Trading Order Architecture

Our order fulfillment is highly extendable, capable of efficiently dealing with a variety of trading order fulfillment (FOK, IOC, etc.). This enables us to provide a professional trading environment for professional traders. This extendable trading order architecture even has the potential to form an ecosystem, available to a wide range of users, including professional financial services providers.

Shard-able Trading Server Cluster

Our trading platform is based on a mature technical architecture with multilayer clustering. It is well-structured to handle high concurrency trading and process a massive amount of information and data.

Meanwhile, we have used extensively Cloud Native compliant frameworks and infrastructures so that our system can be turned into a sharding structure. This has enabled our system to handle a massive trading volume. The maximum trading volume that our system can handle is only dependent on the scale of the server cluster.

Near Real-Time Clearing System

Clearing system is one of the most critical components of a digital asset exchange. We have developed our proprietary asset management system to enable synchronized on-chain and off-chain asset management.

Also, with our multi-layer data architecture, our proprietary clearing system has made near real-time clearing a reality.

Security Architecture

Capital safety and information security of our users are amongst our paramount concerns. Our security architecture includes a separation of hot wallets and cold wallets, multi-signature security, two-factor authentication and others.

2.1.6 Capital Safety and Risk Management

Capital Safety

We have separated our clearing process from our trading platform and have adopted near real-time clearing to ensure the safety of funds on our platform. Meanwhile, we can focus on the optimization of the matching engine of our trading platform.

Account Security

With our account security system based on big data and artificial intelligence (AI), we continuously assess and ensure account security and compliance.

Withdrawal Services

By leveraging big data, AI, and the flexibility offered by human services, our withdrawal services are highly secure, efficient and flexible.

Security Audit

We are working in collaboration with industry-leading cybersecurity companies in the blockchain industry to regularly assess our IT systems.

2.2 Revenue Model

Revenue Sources	Details
Trading Fees	Trading fees will be applicable as a percentage of the value of digital asset transactions.
Withdrawal Fees	Withdrawal fees will be applicable to process withdrawal requests.
Listing Fees	Listing fees will be applicable to list digital assets which meet our digital asset evaluation criteria.
Margin Trading Fees and Interest	We will roll out margin trading services once we have obtained relevant licenses and permissions. Fees and interest will be applicable.
Contract Trading Fees	We will roll out contract trading services once we have obtained relevant licenses and permissions. Fees will be applicable.
Other Revenues	We will leverage our knowledge and expertise in blockchain technology and digital assets to provide other services such as technical advisory, digital asset custodian, and security services. Fees will be applicable.

2.3 Roadmap

Date	Milestones
April, 2018	ABCC.com launched
May, 2018	The first batch of select digital assets listed
July, 2018	ABCC Membership Program launches
January, 2019	New digital asset trading platform launches

3. About Us

ABCC Digital Asset Exchange (ABCC) is a world-class exchange offering digital asset investment and trading solutions for users globally. ABCC was open for registration on April 9th, 2018 and officially launched on April 28th, 2018. Currently, we have dozens of digital assets with BTC, ETH, and USDT as base currencies on our exchange platform. Secure and stable, ABCC has been gaining traction and popularity among users with our offerings of select high-quality digital assets.

3.1 Our Values

Embracing the philosophy of blockchain technology—open, frictionless and participatory, we have placed value investing at the core of our values. Holding investors' interests at the center of our business, we conduct comprehensive due diligence over prospective digital assets before making any listing decisions. At the same time, we work closely with the issuers of digital assets to issue digital assets with great long-term growth potentials.

3.2 Our Competitive Advantages

3.2.1 Industry Leading Technologies

Our technology team members have worked at some of the world's most successful technology and financial services companies. Our technological capabilities have enabled us to create and maintain systems and infrastructures capable of handling matchings for over 10 million users simultaneously. Meanwhile, we are experienced and well-equipped to handle high concurrency trading and process a massive amount of information and data.

3.2.2 Finance Veterans from Top Financial Institutions

Our finance and investment team come from the world's top investment banks and fintech companies, seasoned in designing and operating complex financial instruments.

3.2.3 Global Vision and Experience

ABCC was founded by a group of industry veterans with a broad range of experience and track record at some of the most successful internet, financial services and management consulting firms.

Meanwhile, our international team composition has equipped us with strong capabilities in acquiring and consolidating resources globally. Our team members have previously worked in the US, Europe, the Middle East, Southeast Asia and others.

We are proud to have a number of veterans in legal practices, financial services, blockchain technology and other technologies act as advisors to guide us in our strategic development. Our advisors include but are not limited to

- Dr. Michael Frendo, former Speaker of the Parliament of Malta and Minister of Foreign Affairs of Malta, who has taken an active role in EU Constitutional development and was a signatory to the Draft Constitutional Treaty of the European Convention;
- Mr. Weixing Chen, founder of Kuaidi Dache, Chairman of the Board at Funcity Holding, experienced investor in blockchain technology;
- Mr. Zhang Lei, successful serial entrepreneur who has launched three successful products with each serving over 100 million users;
- Mr. Forrest Chen, founder of Umeng (acquired by Alibaba for \$70 million in 2013) and expert in technologies and strategy.

We thank you for taking the time and effort to learn about ABCC Digital Asset Exchange! If you like what you have read, please join us in helping build a sustainable and healthy blockchain ecosystem!



AB-CHAIN

ADVERTISING NETWORK

BLOCKCHAIN AND AI FOR EFFECTIVE ADVERTISING

We drive traffic for ICO companies
with cryptocurrency budgets

Table of Contents

Terms and Definitions.....	4
Mission and Vision	6
Business Overview.....	7
Problem we solve.....	7
Our Solution.....	8
Our value proposal.....	9
Advantage for Webmasters: Webmasters Wallet	10
Advantages for Advertisers.....	11
Advantages for Advertising Networks.....	11
Market Overview	12
Cryptocurrency Market Value and Growth	12
Cryptocurrency market	14
Market perspective.....	14
Advertising platforms with payment in cryptocurrency	15
Why the conditions of investment dynamics are perfect	16
2017 showed that Blockchain is at its beginning.....	16
Regulated environment	16
Low competition	16
Lack of major players	16
SWOT	17
Competition.....	18
AB-CHAIN competition analysis and a research covering existing advertising platforms that accept bitcoin.	18
AI and ML seen as advantages over big players/rivals.....	20
Our strategy.....	21
Short-term goals.....	21
Long-term goals	21
Product development perspective	21

Webmasters Wallet	22
Advertisers Office	23
Advertising network API	23
Advertising rotation platform	23
Artificial Intelligence	24
How AI works	24
How neural network works	25
Token	27
The name of the Token	27
About RTB Token	27
RTB Token turnover	27
RTB Tokens burning mechanism	28
50% discount for commission	28
RTB Token burning mechanism	29
What happens when all of the RTB Tokens are burned?	29
RTB Tokens purchase by ICO companies	29
Our Plan	30
Crowdfunding structure	31
Funding structure	33
Crowdfunding calendar	34
Pre-Sale and protection of our first investors	34
Main Crowdfunding	34
Protection from cryptocurrency exchange rate	34
Bounty program	35
Team	36
Advisors	38
Sports	40

Terms and Definitions

Webmaster — owner of a website or a blog who earns by displaying ads on his/her website or blog.

Display network — websites or a blogs where ads are placed.

Advertiser — natural person or legal person who bear costs for ad placement at webmasters.

Advertising network — a company that mediates the cooperation between an advertiser and websites wishing displaying ads. The key function of an advertising network is aggregation of advertising places at webmasters and its introduction to advertisers.

ICO (Initial Coin Offering) — an unregulated means allowing fundraising for a new cryptocurrency venture. Startups raise funds through Initial Coin Offering in order to bypass the rigorous and regulated capital-raising process required by venture capitalists or banks.

Fiat — the currency backed by local governments (e.g. USD, Euro, Yen, Pound, Rubble, Renminbi).

RTB Token — the token that will be used to purchase ad placement via AB-CHAIN platform.

Crowdfunding — the practice of funding a project or venture by raising monetary contributions from a large number of people;

Artificial Intelligence, AI — a feature of intellectual systems to perform creative functions that are traditionally considered to be a human beings prerogative;

Machine Learning, ML — Artificial Intelligence subject that studies methods of building self-learning algorithms.

Targeted actions – actions performed by users that are desired by an advertiser, which are goals of ad placement, e.g. registration on advertisers website or its product purchase.

Mission and Vision

AB-CHAINs mission is to deliver the service to companies with cryptocurrency budgets allowing an easy and time-saving way to publish ads on the Internet without converting cryptocurrency into fiat and allowing high returns from advertising budgets.

AB-CHAINs vision is becoming the most technological advertising network that eliminates middlemen in running advertising campaigns by implementing AI and ML¹.

¹ AI and ML – Artificial Intelligence and Machine Learning technologies.

Business Overview

AB-CHAIN is a next generation Advertising Network² that allows Advertisers³ buying advertising places from Webmasters⁴ with cryptocurrency, ensuring high returns from marketing investments.

2017 was the year of ICOs. A significant number of companies that raised funds through ICOs are now developing their products. More than half of these companies aimed at introducing their products in 2018. As soon as these products are introduced, the companies will need to advertise them.

Startups usually spend 20 to 30% of their budgets for marketing and advertising. Budgets for marketing and advertising after initial stage of investment differ from one company to another as introduced below:

- [KICKICO](#) — 43% of investment reserved for “PR & marketing for KICKICO development and project support”;
- [Revain](#) — 45% of investment reserved for marketing;
- [AdEx](#) — 20% of investment reserved for marketing;
- [LordMancer](#) — 50% of investment reserved for marketing.

We expect that in 2018 online advertising market with payments made in cryptocurrency of \$200M will arise, with a yearly growth rate of 120% for the next 5 years.

Problem we solve

Most Webmasters today do not accept cryptocurrency because the demand from traditional businesses is too poor to prove right the new type of payment. In 2018 online advertising market

² Advertising Network – a company that mediates the cooperation between an advertiser and company that mediates the cooperation between an advertiser and websites wishing display ads. The key function of an advertising network is aggregation of advertising places at webmasters and its introduction to advertisers.

³ Advertiser — natural person or legal person who bear costs of ad placement at webmasters.

⁴ Webmaster — owner of a website or a blog who earns by displaying ads.

will experience the emergence of companies that are funded through ICOs and most of it have budgets in cryptocurrency.

These funds will need to be invested. Companies that will convert it into fiat will face a range of problems:

- 1) Low liquidity of cryptocurrency market that causes lower cost rate after conversion;
- 2) Taxes charged by local authorities;
- 3) Trust and reputation loss from the community of investors;

We also see another trend. Traditional companies begin to use cryptocurrency in terms of investment in very small amounts for their everyday needs. Not only our market research but also market leaders state it – Vitalik Buterin, too. This will create a huge market in 5-10 years, which is comparable with the one arising through ICOs fundings.

Our Solution

AB-CHAIN will provide companies running their ICO as well as those that already raised funds through ICO, and other businesses with cryptocurrency budgets with an advertising network. There they are able to buy online ad using cryptocurrency (RTB token), virtually eliminating the need to convert it into fiat.

We will be able to satisfy the demand for any purchase amount of advertisement by:

- Our own traffic and direct agreements with website publishers
- Direct connection of third-party ad networks to the platform AB-CHAIN
- Linking-up traditional publishers to our platform that will allow them to add bitcoin as a payment method.

AB-CHAIN is working for development of a sustainable advantage over traditional advertising networks not only through the adoption of cryptocurrency, but also by increasing the effectiveness of advertising placement. AB-CHAIN uses AI and ML to automatically maximizing the desired actions from the posted ads.

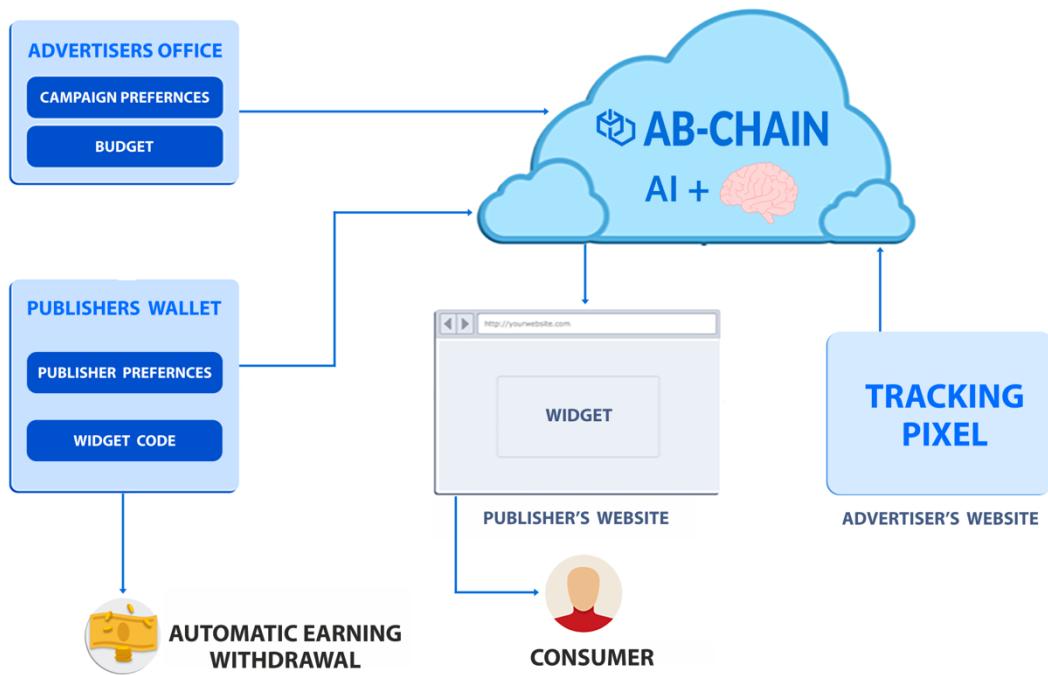
That will allow you to:

- remove the middlemen functioning as a filter of ineffective marketing channels
- dispose of fees for intermediary's services, which usually ranges from 15% to 30%
- eliminate the human factor in the determination of placements and increase the efficiency and speed of channels' selection
- work with business metrics (ROI and ROMI), which are generally either not tracked, or controlled "on a hunch".

Our value proposal

AB-CHAIN eliminates the middlemen between advertisers and websites that host advertising, and also creates the universal payment of ad placements by introducing your RTB token.

Within AB-CHAIN we want to provide benefits to all participants of the advertising process through our platform: webmasters, advertisers, advertising networks.



Advantage for Webmasters: Webmasters Wallet

Web-publishers traditionally do not accept payment by cryptocurrency. However, advertising spaces usually sell less than 50% of the ad capacity, and then they sell what remains at the lowest prices. . We believe that most webmasters will also be ready to accept payment in cryptocurrency, if provided with the tools that allow it.

We decided to create a wallet that allows webmasters to sell advertising space with greater efficiency and profit.

The wallet gives you following possibilities:

- 1) It gives the opportunity to receive payment in cryptocurrency;
- 2) It allows webmasters to withdraw fiat money if necessary.

Additional benefits for all website publishers:

- 1) Webmasters do not need to switch advertising widgets of different networks, it is enough to put the widget AB-CHAIN, which will fill the entire advertising capacity;
- 2) Universal AB-CHAIN widget with the best market practices;
- 3) Automatic transfer of webmasters' earnings on their crypto-addresses.

Advantages for Advertisers

AB-CHAIN plans to insure the benefits for advertisers through:

- 1) Providing advertisers with a wide range of Webmasters that propose effective advertising opportunities;
- 2) Providing advertisers with the opportunity to make payments with cryptocurrency;
- 3) Providing them with display opportunities at webmasters who normally do not accept payments with cryptocurrency;
- 4) Eliminating the need to exchange cryptocurrency for fiat;
- 5) Automatically optimizing targeted conversion data via AI and ML technologies.

Advantages for Advertising Networks

AB-CHAIN plans to provide a solution for existing advertising networks:

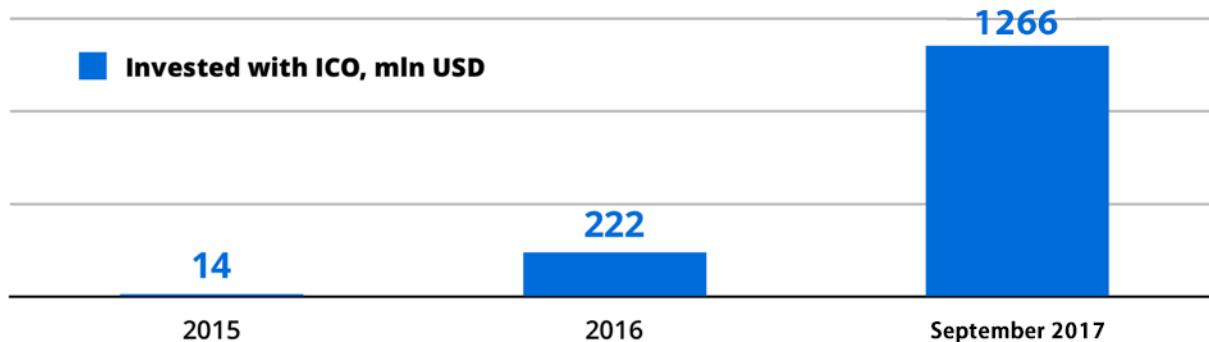
- 1) By introducing the opportunity to increase the sales with ICO advertisers. It will be real with API covering the need to import advertising banners uploaded into display network in order to display it at webmasters via AB-CHAIN network;
- 2) By allowing payments with fiat and cryptocurrency.

Market Overview

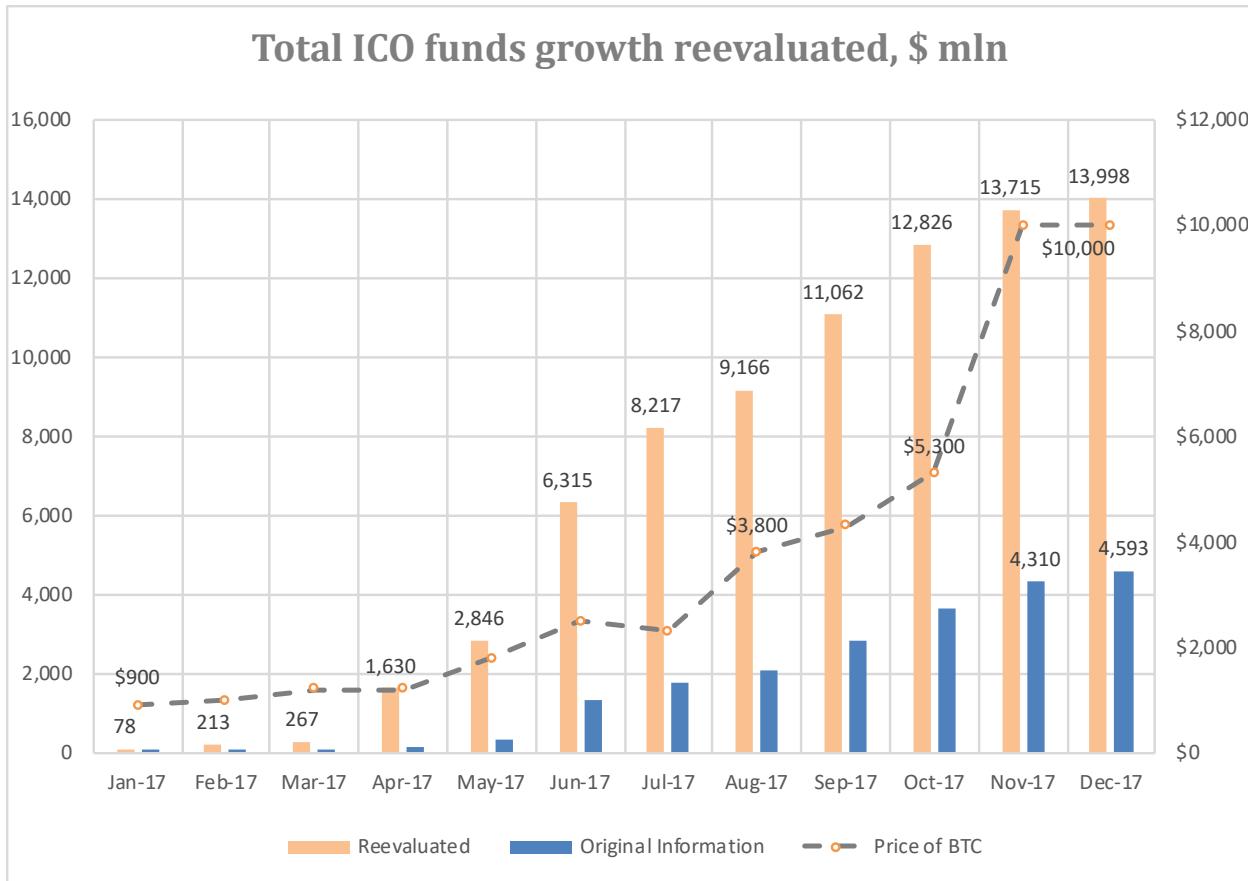
Cryptocurrency Market Value and Growth

ICO funding [grew](#) exponentially in 2017.

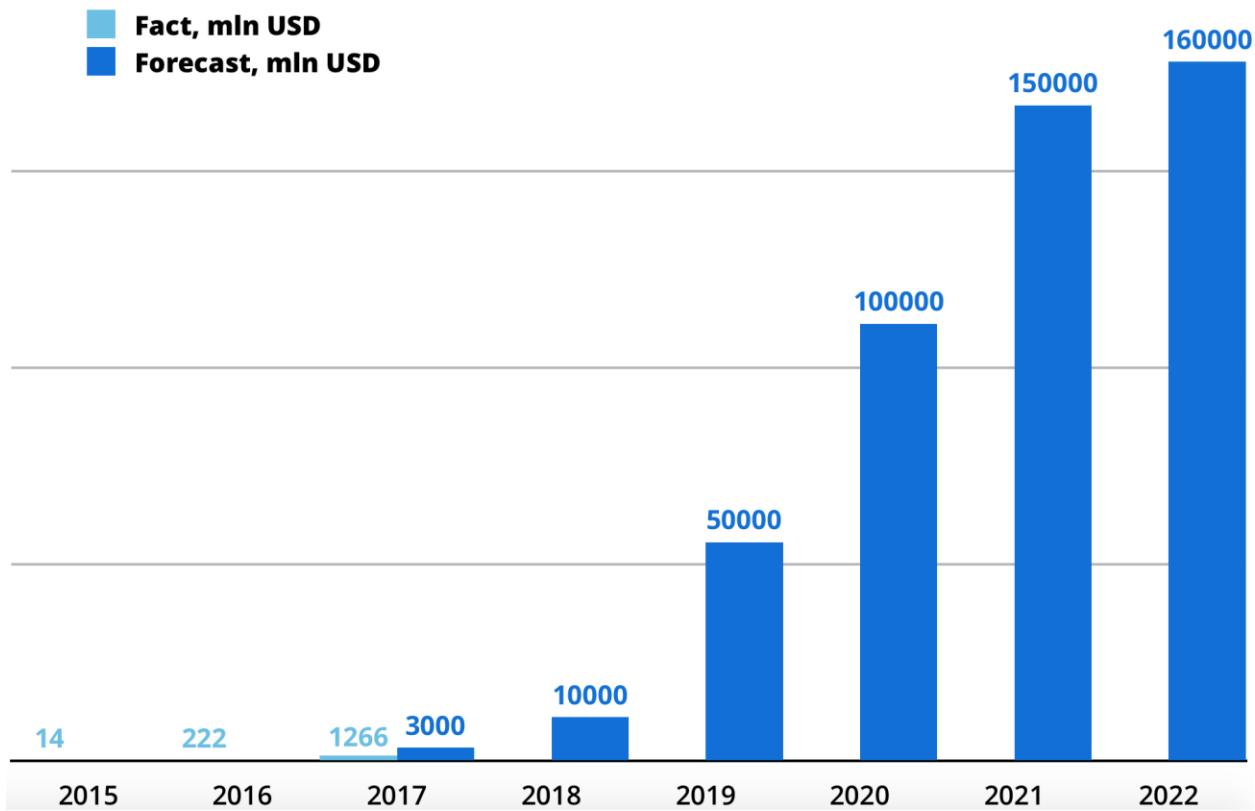
August 2017 showed [\\$1.9 bln](#) funded through ICOs which means that the number increased by \$300M after SEC (Securities and Exchange Commission) issued its report on ICOs and after [46 new Coin Offerings have been announced and an additional 2014 are moving toward fundraising.](#)



The second Quarter of 2017 market growth went on. In September 2017, AB-CHAINs analytical department predicted the total amount of funds raised through ICOs in 2017 will be \$3bln. However, nominal statistics shows \$4.5bln; taking the growth of cryptocurrency rate into account, the total amount is \$10bln (as of January 2018).



The exponential growth of ICO funding reflects a significant potential of this market and is caused by the so-called “low base effect”. We expect that until 2022 the growth of ICO market will be \$120bln yearly, which means annual increase of 30 times compared to 2017.



Cryptocurrency market

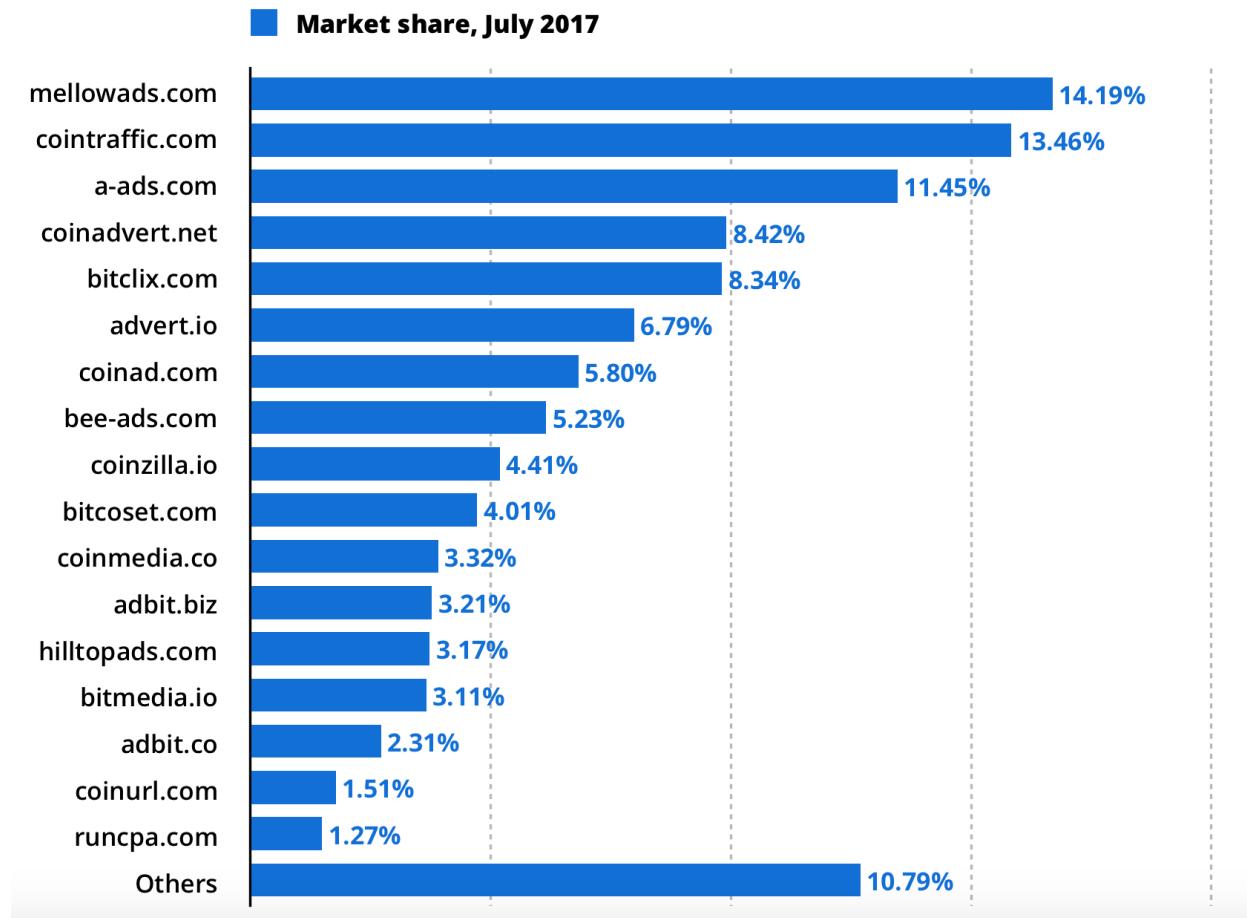
According to the Global Benchmarking Study based on [interviews](#) with 48 companies that accept and make payments in cryptocurrency in 27 countries, 79% of these companies cooperate with banking and payment systems. Yet, this rate is insignificant comparing to traditional payment market such as international payment systems (Visa, MasterCard, UnionPay etc.) or mobile wallets and other payment methods.

Market perspective

Although it is hard to predict the market growth, [Saxo Bank](#) sees Bitcoin can take up to 10% of the \$5 trillion daily foreign exchange market in 10 years. Its market capitalization can grow up to \$1.76 trillion, which corresponds to the cost of Bitcoin \$100 000.

Advertising platforms with payment in cryptocurrency

Our marketing studies show that most of the companies, which accept payments in cryptocurrency, are aimed at anonymous payments market share.



This market is quite insignificant with less than \$10M annual turnover. We expect that the future leaders will be able to deliver their services both for companies funded through ICOs and Webmasters accepting fiat. Currently both do not have access to the services.

Why the conditions of investment dynamics are perfect

2017 showed that Blockchain is at its beginning

The market is supported by investments made with cryptocurrencies. The adoption of technology is ubiquitous, including several large companies, banks and funds that are working or investing in Blockchain projects. IBM has more than 200 Blockchain projects that are being developed with corporate units. Large international banks also [begin](#) working on their Blockchain projects.

Regulated environment

A serious obstacle in the past, regulation becomes a positive force for the growth of cryptocurrency today. The Japanese government officially recognized Bitcoin in April 2017, giving it greater legitimacy in one of Asia's richest economies.

Countries actively working on regulation - Singapore, Switzerland, Russia, Estonia, etc. It is expected that soon other countries will follow, and as a result, the constant growth of international cryptocurrency transactions.

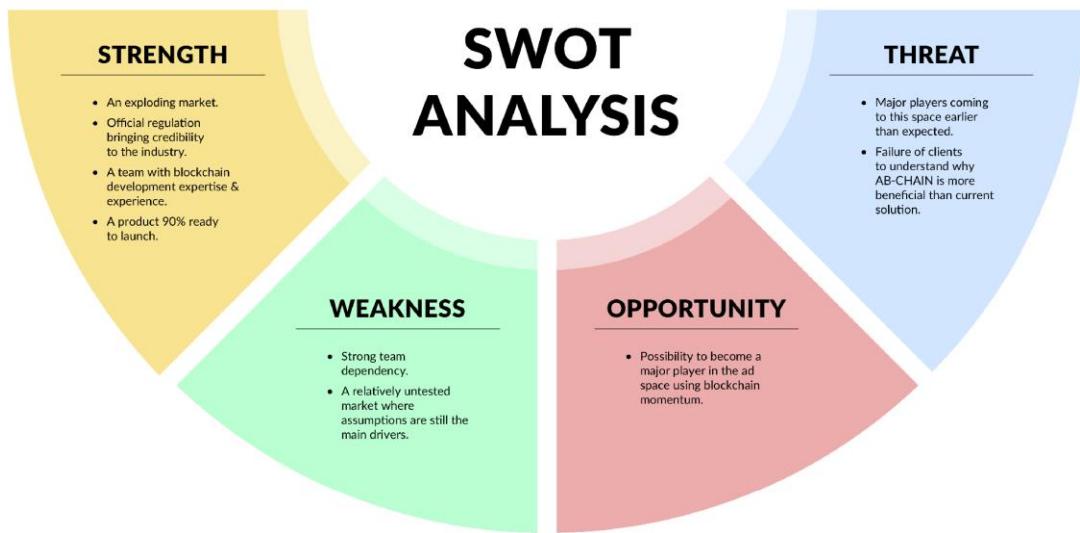
Low competition

The number of services and startups that take this market opportunity is increasing. However, the number of players is still relatively small, which leaves free space for newbies. Since competition is limited, the market share is sufficient to minimize risks. Cryptocurrency market is an advantage for new services.

Lack of major players

The market currently lacks on major players despite expectation of companies in free market space. We expect the grace period will last 1 to 3 years and our company has enough time to expand its competitive advantage.

SWOT



Competition

We will be competing with lots of companies both directly and indirectly. Our indirect rivals are traditional advertising networks, e.g. BuySellAds.com, Adsterra.com, Adcash.com and so on, as well as other types of services that propose advertising services for cryptocurrency, fiat, cash.

Our direct rivals are traditional advertising networks. Key differences of AB-CHAIN:

- 1) AB-CHAIN accepts cryptocurrency for payment;
- 2) AB-CHAIN has expertise and experience in Blockchain technology unlike traditional advertising companies;
- 3) AB-CHAIN aggregates other networks to ensure fast growth. This will allow us reach suitable business scale to compete with existing industry leaders;
- 4) AB-CHAIN implements AI and ML technologies in order to have advantage over market players both crypto and traditional networks.

AB-CHAIN competition analysis and a research covering existing advertising platforms that accept bitcoin.

Existing advertising networks that accept payment in cryptocurrency are mainly concentrated on two markets:

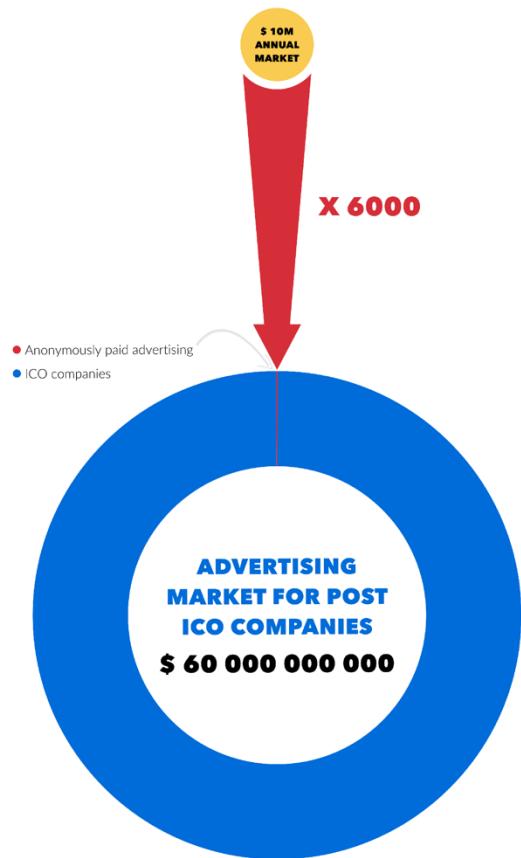
- 1) Companies that raised funds through ICOs:

This is a “white” and legal market with legal business and startups. The innovation and technology are fully legal so such businesses do not look for any kind of weaknesses in local jurisdictions. The market will grow 6 000 times comparing to the dark advertising market (with payments made anonymously) as expected.

- 2) Companies that compose the dark market:

Such companies literally buy an opportunity to make their payments for advertisement anonymously. Great examples are online casinos and adult entertainment industry. Even though these services are easily accessible it is illegal in almost every jurisdiction. The payment in cryptocurrency is their last chance to deliver the product.

Anonymous payments for advertising services has grown up to the maximum. We expect 10 to 20% of yearly growth rate.



AI and ML seen as advantages over big players/rivals

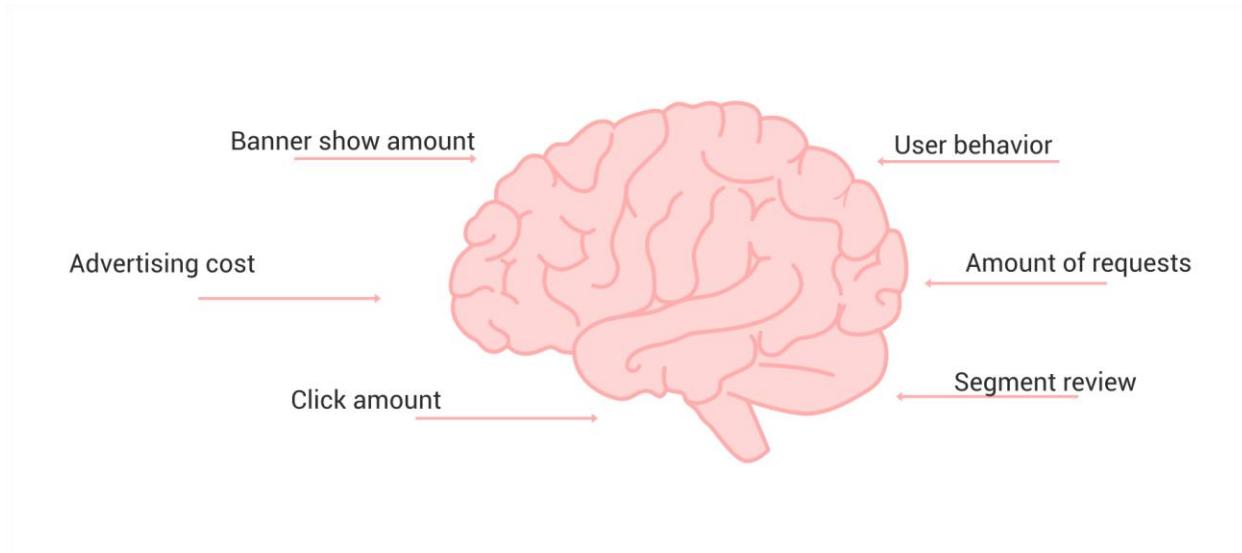
Except for holding great market shares, big players also solve advertisers problems which leads to attracting clients. This concerns Google, ApNexus etc

In order to hold a significant market share we need to introduce an even more attractive solution.

AB-CHAIN is developing a solution to maximize the return of investment in advertising. In this case, we implement AI and ML technologies.

Google and Yandex already use these kinds of solutions. The solution is a system of automatic optimization of CTR, that is, the conversion of ad views into visits. Actually advertisers goal is to maximize the investment in advertising into targeted actions (registration or purchase). Existing market players do not solve this problem delivering only the intermediate solution.

AB-CHAIN is approaching the Advertiser by helping to maximize the target result and not the intermediate (clicks).



Our strategy

Short-term goals

Our short-term goal is to sign contracts with at least 10 ICO projects for long-term services supply. This will prove market demand for our product before we proceed with expanding geographical coverage.

Long-term goals

Our long-term goal is to grasp a significant market share in online advertising with payments made in both cryptocurrency and fiat.

We will achieve it by first, becoming a leader platform that accepts payments in cryptocurrency delivering our services to ICO projects and to traditional business that will gradually switch to cryptocurrency in making payments for outsourced services which is online advertising.

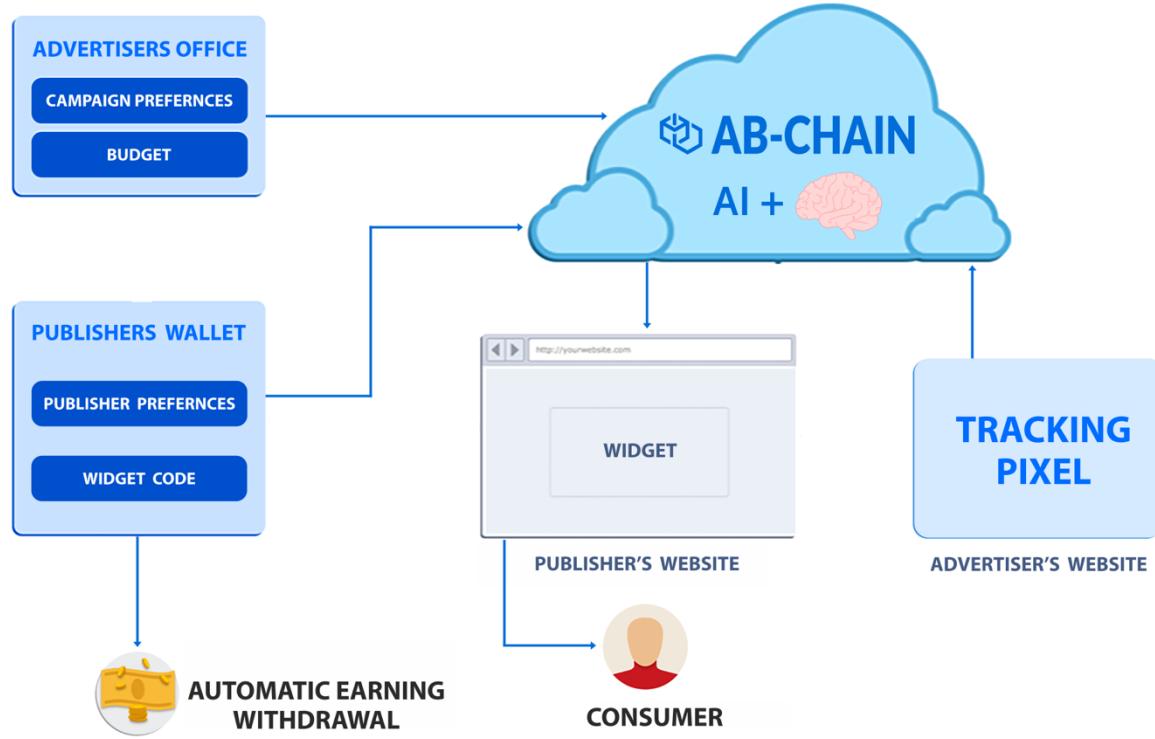
Our next step is to attract companies that are not aiming at making payments in cryptocurrency. We are developing a sustainable advantage by automatic optimization of advertising goals provided by AI and ML technologies.

Product development perspective

AB-CHAIN platform includes several components that interact with each other:

- 1) Webmasters Wallet
- 2) Advertisers Office
- 3) Advertising network API
- 4) Advertising rotation platform

Integration with cryptocurrency stock exchanges for accepting and making payments in fiat.



Webmasters Wallet

Webmasters Wallet automatically accepts RTB Token for ad placement. Webmaster is able to exchange RTB Tokens into any cryptocurrency via stock exchange.

Webmasters Wallet key functions are:

1. Wide range of settings for a Webmaster such as ad categories acceptable for display, banner sizes and types (static pictures, videos, GIF-files etc.), device type and so on;
2. Widget settings interface;
3. Payout settings;
4. Advertising statistics;
5. Payment history.

Advertisers Office

Advertisers office will include following functions:

1. User friendly interface;
2. Possibility to pay with RTB token – AB-CHAIN platforms token – as well as with top cryptocurrencies via stock exchanges;
3. An easy-to-use wizard for creating and editing advertising campaigns that allows to put all the necessary targetings;
4. Analytics and conversion analysis module;
5. An option to choose any Webmaster from AB-CHAINs partner networks.

Advertising network API

AB-CHAIN introduces a solution to boost sales for existing advertising networks. This will be brought by RESTful API including

1. Publishers list synchronization;
2. Postbacks allowing both publishers and advertisers to initiate campaigns ending;
3. Conversion and analytics reporting;
4. Payment reporting;
5. Payment preferences.

Advertising rotation platform

Advertising rotation platform is our businesses core technology that allows automatic ad views monitoring with offer changes real-time tracking. This constant monitoring allows the system to change ads for each specific Webmaster automatically and produces ad rotation. This platform delivers the concept of Real Time Bidding, the real time bidding for ad places.

Our advertising rotation system is a cloud-based application scaling, depending on the load of the network. This allows the system to scale for a big volume of advertising without changing the software.

AB-CHAIN advertising rotation system implements AI and ML technologies in order to boost the return from advertising budgets.

Artificial Intelligence

AB-CHAIN introduces the solution for maximizing the conversion of advertising campaigns into targeted actions, and due to it differs from existing traditional market players.

Challenge for us is to introduce our solution as global taking into account that each advertiser and each campaign are unique. Below are some of the factors that affect advertising campaigns:

1. The product being advertised and messages being displayed;
2. Unexpected reaction of the audience on each new advertising campaign, its banners and texts;
3. Different target audience;
4. Time and days of the advertising campaign;
5. The change in audiences attitude to the brand;
6. Others.

AB-CHAIN implements Artificial Intelligence and Machine Learning technologies as a global solution.

We use a self-learning neural network that is able to analyze advertising campaigns in real time and that connects the results of ad placement with its conditions.

How AI works

AB-CHAIN implements AI efficiently:

- 1) We accept elements of corporate identity for advertising messages;
- 2) Places a targeted action tracking program on Advertisers website – the so-called Tracking Pixel;

AI automatically marks ad placements in order to verify banners, visits and targeted actions and to identify which placements lead to targeted actions . It automatically combines advertising material in order to identify those generating targeted result. It also tracks other variables such as websites, audience interests, time etc.. These are tests initiated by AI.

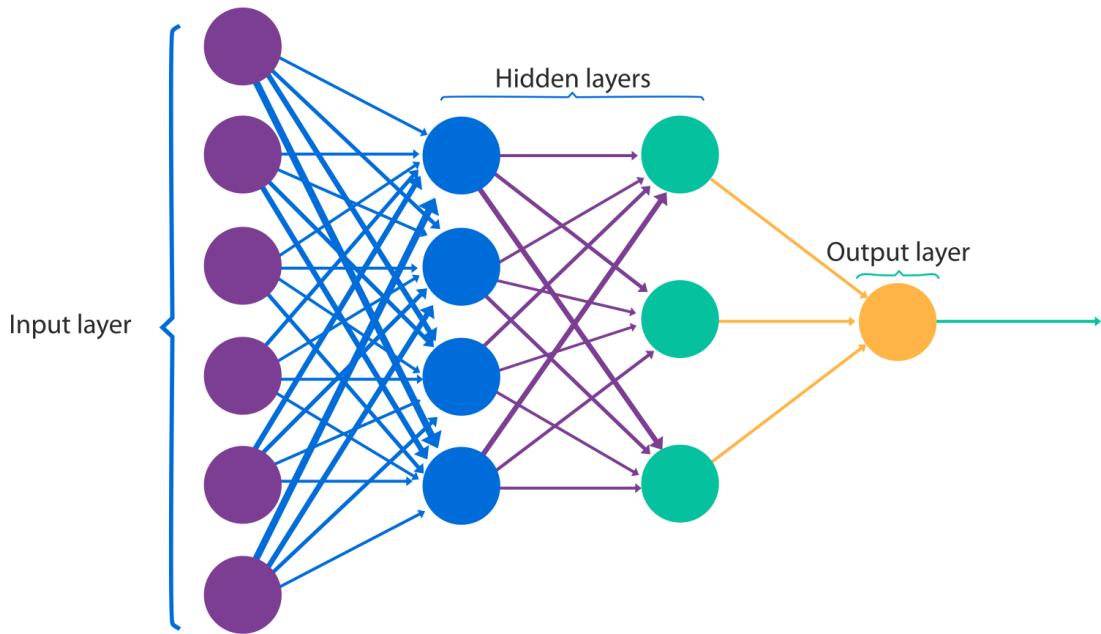
During the display AI is monitoring statistics and performs Machine Learning:

- 1) Placement is made;
- 2) The result is ranged as high and low;
- 3) Machine Learning receives the data on placement factors that came along with high results which stimulates positive samples;
- 4) It receives data on low results as negative samples;
- 5) Placement;
- 6) Control of results improvement;
- 7) Integration repeat.

How neural network works

Neural network is a mathematical model along with its programmed or hardware implementation which is build on biological neural network concept, organization and functional of the cell network of a living organism.

A neural network is able to self-learn in its non-algorithmic nature. The core point of the network is that a set of signals is fed to the input, which passes through the network (neurons with their mathematical functions and trained variables), and generates a response at the output. That is, the signal is applied to one neuron, it calculates the response, passes it to the next one. The next one receives it from the previous one and its neighbor and even more complex combinations are possible. Next neuron calculates its function in terms of its coefficients. And so on until the end of the network where it gives out the result.



The process of self-learning performed by the network is required for the adjustment of its coefficients with the set “given-expected” or “given-not expected” (positive and negative samples). The network learns how to give the right answers, and how not to give false answers.

The process of self-learning performed by the neural network is iterative. Each “lesson” requires the next iteration to give a better response.

This whole process requires the network to identify the rules not through an algorithm but through an experience, which means through interacting with the environment.

The meaning of this is that the network as it finds a regularity, but not algorithmically, but experimentally, that is, interacting with the environment. This allows achieving higher results in unpredictable conditions that contain regularities comparing with algorithms.

Token

The name of the Token

AB-CHAIN is the issuer of RTB Tokens.

RTB stands for Real Time Bidding. This means bidding occurring in real time – the technology in online advertisement industry that is based on IAB OpenRTB protocol.

We have chosen this name for AB-CHAIN Token to specify our professionalism and expertise in online advertising.

About RTB Token

RTB Token is developed as a decentralized ERC20 Token based on Ethereum blockchain. This allows AB-CHAIN to use smart contracts to cooperate with its clients.

In order to boost RTB Token adoption we suggest our partner networks adding it as payment currency.

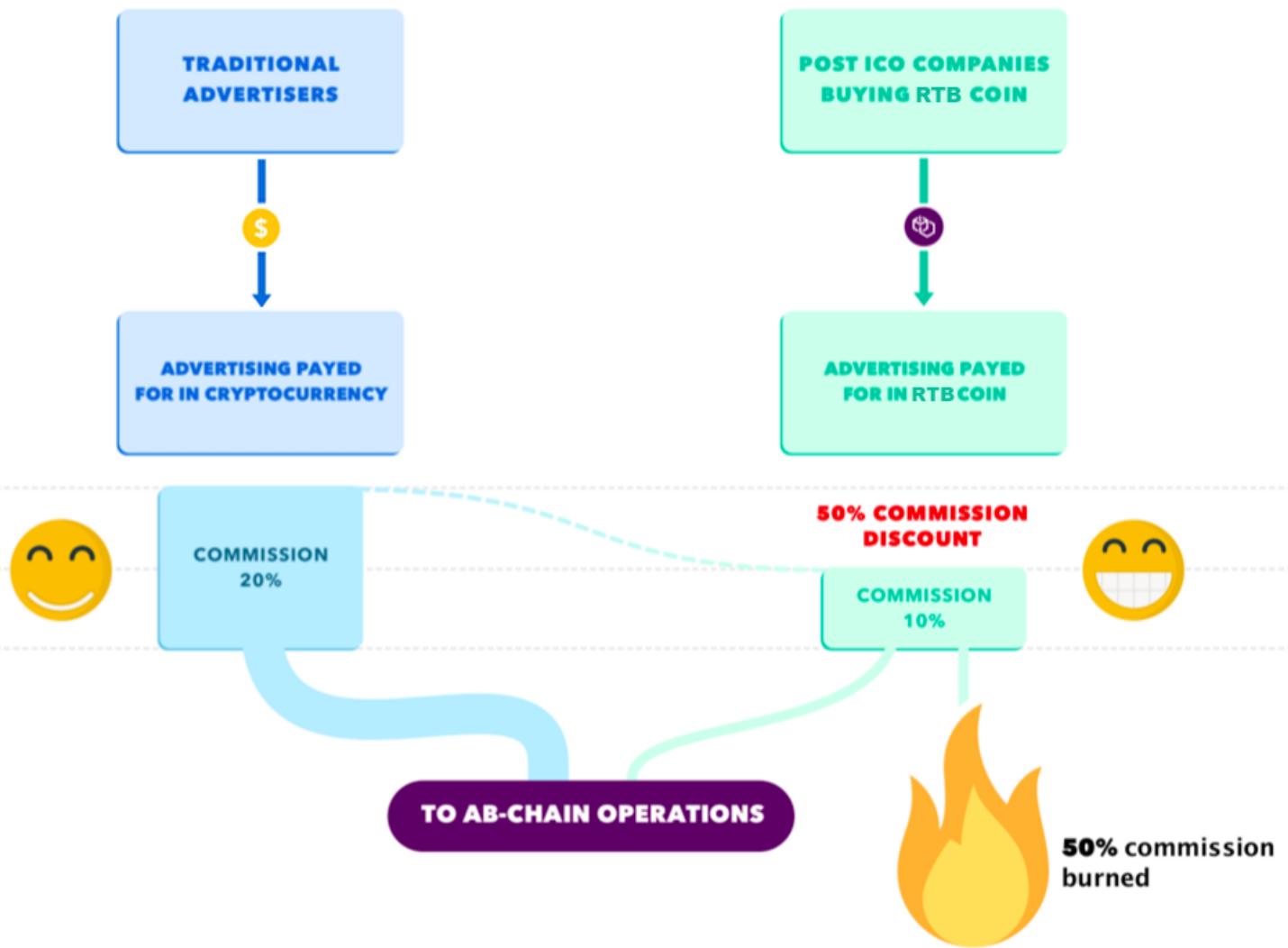
RTB Token turnover

AB-CHAIN requires commission of 10% for each transaction made with RTB Token.

50% of the commission is burned and the other 50% is used for AB-CHAIN blockchain supply which means AB-CHAIN compensates its operating expenses and marketing activities.

RTB Tokens burning is the contribution to AB-CHAINs advertisers, webmasters and investors community.

RTB Tokens burning mechanism



50% discount for commission

The main incentive for advertisers to make payments in RTB Tokens is to get a 50% discount on the AB-CHAIN commission. Advertising networks charge an average of 20% of the commission for all placements. An advertiser paying for placement with fiat will be charged with 20% commission while an advertiser paying with RTB Token will be charged with only 10% commission.

RTB Token burning mechanism

In order to stimulate interest for RTB Token we introduce the burning mechanism. We burn 50% of the commission for contribution to AB-CHAINs advertisers, webmasters and investors community.

An advertiser who purchases ad display for 100 RTB Tokens will be charged with the commission of 10%. Half of the commission will be burned and the other half will be used for covering the expenses of the AB-CHAIN network (see [RTB Tokens burning mechanism sheme](#)).

What happens when all of the RTB Tokens are burned?

RTB Token has 18 токен и имеет 18 decimal characters and can be divided as any other cryptocurrency. We expect that the network will have enough RTB Tokens to function within the next 5 years. When RTB Tokens supply will bypass the point where failures in services delivery are possible due to the small amount of RTB Tokens, then the new token will be introduced along with the possibility to exchange remaining RTB Tokens

RTB Tokens purchase by ICO companies

We target advertisers who raise funds through the ICO as our target partners. The average time between a successful ICO and the release of a product is at least 6 months. That is why we welcome these companies to purchase RTB Tokens at an early stage, allowing them to immediately advertise their ICO and to promote their products for RTB Token in the future.

We will cooperate with the companies that are currently working on their ICOs or successfully completed it.

Our Plan

- 1) 2017 Q4 – The release of the AB-CHAIN platform for its first advertisers (ICO projects)
AB-CHAIN launched its advertising network and attracts webmasters to supply ad placement for its first advertisers. The functional allows publishers to install widgets to their websites. It allows advertisers to set personal accounts. Advertising campaign reporting and moderation are set. AB-CHAIN network is able to provide 500 000 impressions daily and successfully launched first advertising campaigns.
- 2) 2018 Q1 – RTB Token establishment and token burning mechanism
Our Token is already issued. It is a smart-contract in Ethereum network based on ERC20 standard. RTB Token maintains the function of burning along with the mechanism of Token migration into a different smart-contract in case of a failure or in case there is too little amount of Tokens left.
- 3) 2018 Q2 – Smart-contract for advertising deals
The smart contract will allow transactions between traffic sources (publishers and advertising networks) and advertisers. The commission of 10% for the advertising network AB-CHAIN along with 50% burning due to contribution to the community of advertisers, publishers and investors will turn AB-CHAINs proposal to advantage among rival proposals.
- 4) 2018 Q3 – AI development
Creating an artificial neural network will allow analysing the development of advertising campaign online and will establish correlation between the results and the conditions of placement.
- 5) 2018 Q4 – Grasp the market of ads paid with crypto
AB-CHAIN will be scalable through connecting new traffic sources as well as marketing and PR campaigns aiming at attracting new advertisers. Along with it will come constant enhancement of users interface (advertisers office) for a convenient creating, evaluating and managing advertising campaigns.
- 6) 2019 – Grasp traditional digital advertising market
The number of those companies implementing blockchain and cryptocurrency is obviously growing and affecting the traditional market. This trend will be increasing in the future. AI and ML technology will allow us develop privilege which will lead to constant growth at the traditional digital market as well

Crowdfunding structure

RTB Token is developed as a decentralized ERC20 Token based on the Etherium Blockchain. The Token will be issued . 100 000 000 RTB Tokens are issued.

We are raising funds in two stages: pre-sale and crowdfunding (main sale). Funds raised during the pre-sale are distributed for organising crowdfunding, hiring first employees, platform development.

The Tokens will be introduced and distributed as follows:

	PRE SALE	Crowdfunding
Token number	5,000,000 (or 5%)	68 500 000 (or 68.5%)
Price / Token	\$ 0.20	\$ 0.35
USD Goal	\$ 300 000	\$ 1 500 000
USD max	\$ 500 000	\$ 20 000 000

All unsold tokens will be destroyed after the crowdfunding is complete. Tokens burning will appear in each category (team, employees, etc.) to ensure that investors / buyers of tokens during crowdfunding have received 70% of RTB tokens in total.

The bonuses during crowdfunding will depend on the total amount of funds raised and will be reducing gradually from 35% to 10%:

Stage	Bonus
Private Sale	35%

0 - \$ 2 000 000 30%

\$ 2 000 000 - \$ 5 000 25%

\$ 5 000 000 - \$ 10 000 20%

\$ 10 000 000 - \$ 15 000 15%

\$ 15 000 000 - \$ 20 000 10%

Tokens distribution:

Founders 15 000 000 (or 15%)

Employee 5 000 000 (or 5%)

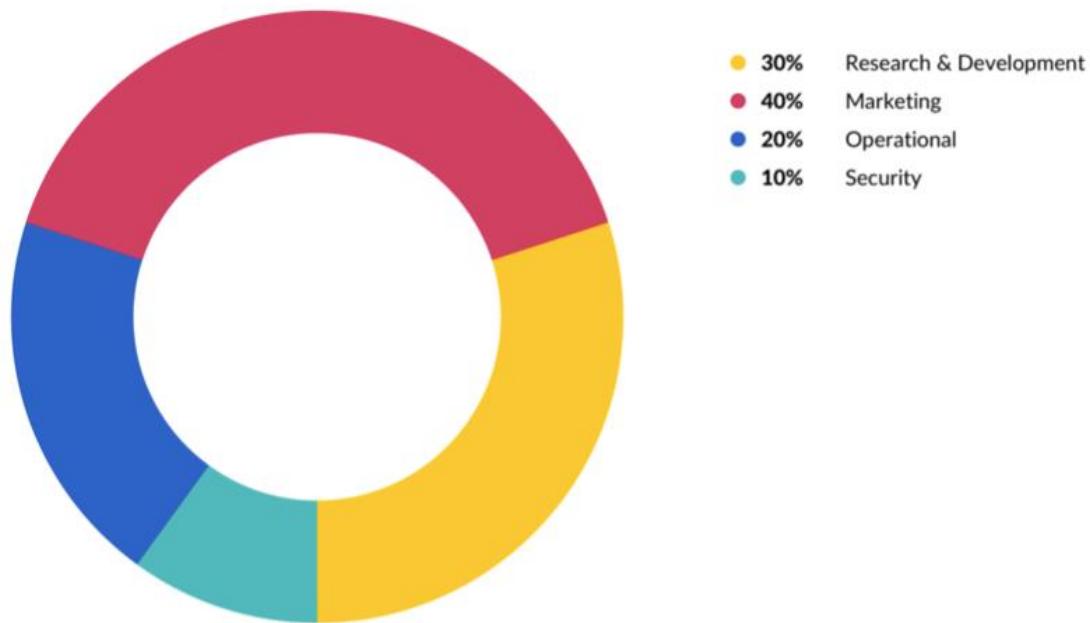
Company reserve 4 000 000 (or 4%)

External consultant 3 000 000 (or 3%)

Bounty hunting 3 000 000 (or 3%)

Funding structure

- 1) The planned use of funds received during Crowdfunding:
 - 30% of the funds will be used on R&D
 - Software development
 - Open Source Development investments
 - Block and crypto currency Development investments
- 2) 40% of the funds will be used for marketing
 - Sales offices: Asia, Europe, America
 - Incentives and events for webmasters
 - Sales managers motivating program
- 3) 20% of the funds will be used for operating expenses
 - Offices rent
 - Legal and accounting services
 - Server infrastructure and cloud services
- 4) 10% of the funds will be used to secure the platform



Crowdfunding calendar

AB-CHAIN Crowdfunding is carried out in two stages.

Pre-Sale and protection of our first investors

AB-CHAIN successfully held a pre-Sale, which ended on October 1, 2017, having collected \$ 370,000.

It was originally planned to hold AB-CHAIN Crowdfunding at the end of 2017, but we moved it to February - March 2018 due to the first version of the platform unpreparedness. To all Pre-Sale investors we accrued an additional bonus of 30% for expectation.

Main Crowdfunding

The Main Crowdfunding will be held from February 19 to March 31, 2018. There will be sold 68,500,000 RTB tokens maximum:

- The minimal funding is \$ 1,500,000.
- The maximal funding is US \$ 20,000,000.
- The price of the RTB token is \$ 0.35.
- Bonuses will be distributed from 10% to 30%.
- Additional bonuses for large investors (from \$100k).

* The minimum deposit amount is set to 0.2 ETH or 0.015 BTC.

Protection from cryptocurrency exchange rate

Protection of investors against the growth of the rate of the crypto currency

Due to the growth of bitcoin and ether since September 2017, we collected a large amount of feedback from our early investors regarding the fact that they received significantly fewer tokens than later investors, despite a higher bonus. To solve this problem and protect Crowd-Funding investors from a significant increase in the cost of the crypto-currencies invested in AB-CHAIN, we have introduced a number of new rules.

We will recalculate the number of AB-CHAIN tokens for all investors who bought AB-CHAIN tokens before the start of Pre-Sale and Private Sale a day before the start of Crowdfunding (February 18), based on the maximum rate of the two dates (the investment date and February 18th). For example, if the investor invested in Pre-Sale 10 ETH at \$400 in October 2017, and on February 18 the exchange rate will be \$1200 for 1 ETH, all bonuses will be preserved, and the number of tokens accrued to the investor will increase by 3 times, due to the growth rate of 3 times .

The base price of the token for the whole of Crowdfunding will be in USD and will be \$ 0.35. On February 18, the price of the AB-CHAIN token will be fixed at the rate of BTC and ETH to the dollar for the entire Crowdfunding.

If the exchange rate changes during Crowdfunding by more than 20%, the price of the token may be changed in accordance with the change in the exchange rate to preserve the interest of new investors.

The last day of Crowdfunding, March 31, we will recalculate and present to each investor the maximum rate based on the rate of the two dates - the moment of investment and the time of Crowdfunding completion (in case of growth).

Bounty program

We offer the participants of the AB-CHAIN bounty program a total distribution of 3% of the tokens amount. The total number of tokens in circulation may vary (depending on the actual amount raised during Crowdfunding, the maximum goal is \$ 20,000,000), but in any case 3% of the tokens will be directed to the Bounty program.

Team

AB-CHAIN is being developed by an experienced team of developers and business professionals. We have already successfully delivered multiple complex project such as:

PINbonus - A programmable electronic card with its own iPhone/Android application that replaces all discount and reward plastic cards (barcode, number, picture, magnetic stripe)

QIWI Bonus - An ad platform both for consumer goods and financial traffic for QIWI (NASD:QIWI) and other publishers. Additionally we also created a cashback service <https://bonus.qiwi.com>.



Vladimir Dyakov

CEO, Founder

15 years in development

10 years in management

7 years in sales рекламные сети и финансовые сервисы

4 years in Blockchain; several successful projects, among investors are FINAM Global, NASD:QIWI, Prostor Capital



Ivan Pshenitsyn

Co-Founder, CTO

11 years in development

5 years in management

ad networks and financial services

6 years in Blockchain



David Pomies

Co-Founder, Director of Development

9 years in sales and marketing, 7 years in business development in Hong-Kong



Cate Lawrence

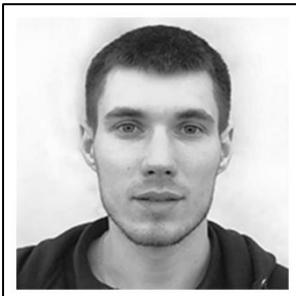
PR

4 years in technology journalism in Berlin

7 years in charity in Australia

10 years in higher education

Published in [VentureBeat](#), [Sitepoint](#), [DZone](#), [ReadWrite](#),
[TheNextWeb](#)



Antony Oshkin

Chief Marketing Officer

Founder at RocketLP Digital Agency

6 years in digital marketing

4 years in marketing development

More than 50 successful marketing strategies



Adrien Henni

International strategy advisor



Dom Inzerillo

International Marketing and Community manager



Sergey Kupryanov

Developer



Alexey Shalin

Systems Administrator



Ivan Rogozhev

Developer



Slava Mirgorod

Developer



Kirill Remizov

QA Ingeneer



Veronika Repyeva

Manager



Ivan Skladchenkov

Community Manager



Anna Mandryuk

Community manager

Advisors



Reuben Godfrey

ICObench Expert (TOP-10)



Juan Garay

Texas University A & M



Arseny Strizhenok

ICO consulting
LAToken, Blockchain &
ICO consulting
EWDN.com



Kaimin Hu

AI Product manager at Advance.ai



Kirill Ermakov

Technical Director,
QIWI



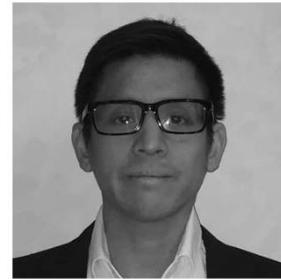
Igor Bulatenko

Security Director, QIWI



Christos Stergiou

Connection to investors
in Greece and Canada



Victor Ho

Connection to investors
in China



Leonid Delytsyn

Chief Analyst, Finam
Global



Artem Ozerkov

Head at Webmasters
Accounting Department
at admitad.com



Alex Savchenkov

Founder and President
at cityads.com



Dmitry Kozlov

Head at Product
Department at Alfa
Bank

Sports

When not busy with our work, the AB-CHAIN team is also actively practicing sports. While Vladimir and Ivan specialize in running mountain events (see their last event below) David is running for pleasure in local events.

ADIDAS ELBRUS WORLD RACE 2017 - 11 KM AND 34 KM RUN

Elbrus is the highest peak of Europe that consists of 2 peaks: the Western 5642m and the Eastern 5621m (31 meters lower). The mountain is located in pictorial Baksan gorge where each year since 2012 international mountain trail running competition takes place.

Our co-founders, Ivan and Vladimir, took part in the run in 2017, Ivan has finished 11km distance with +1800m of altitude gain and Vladimir has finished 34km distance with +2700m up. We are happy to share some pictures from the run with you.



MARATHON DU MONT BLANC 2017 - 42 KM



Last year, Vladimir has finished one the most beautiful marathon runs worldwide - Le Marathon du Mont Blanc. The race starts in Chamonix and drives runners through the gorgeous Alps mountain landscape.

FRENCH NATIONAL ATHLETE



David is a Former French National Athlete and former Oklahoma State University cross-country runner. His personal records are impressive 8.04 on 3000m, 14.07 on 5000m, even though he retired a while ago, he still enjoys participating to local races in Hong Kong and Europe.



Absolute. Proof of View

**Litepaper
V1.1**

10-June-18

This Litepaper is a cut down version of the whitepaper which is still in development. This paper assumes that you already have a basic understanding of the cryptocurrency environment.

The Coins Vision:

Our vision is to provide an advertising, promotions and event platform with smart contracts that will have a global reach of network enablers. Our network will allow agents, organisers and promoters the ability to reach a global audience and utilise their participation to achieve their desired media goals.

The network is driven by ultra-low transaction fees, reliable and fast transactions (10x faster than Bitcoin) which are supported by a large core network of Masternodes.

Mainnet Development

A newly developed Mainnet (specifications of which will be announced shortly) will be attached to the current blockchain. De-synchronous sidechain implementation for smart contracts will be initiated via collateral agent inputs and managed under escrow from upper tier nodes.

Reward Structure

Proof of Stake (POS), Proof of View (POV), Proof of Work (POW), Masternodes (MN)

The ABS Cycle

Absolute aims to incentivise a healthy balance of holding and liquidity. To do this we will split the reward structure as follows:

Holding Incentives

- **Masternodes**
Masternodes are a way of earning rewards for holding coins: collateral is locked into the network allowing stability. In return, Masternodes take it in turn to process the network, receiving a reward for doing so.
- **POS**
If an individual cannot hold enough coins to activate a Masternode, they can earn rewards by holding over a specified amount until the coin matures. When this happens they will start to receive rewards in the form of ABS.

Liquidity Incentives

- **POW**
Hashing power from mining will supply security and stability to the network. In return the network will offer 10% of the block chain reward structure to incentivise liquidity.
- **POV**
The Proof of View network is funded via smart contracts. Agents set contracts which are paid per interaction with the network. The interaction rewards are set by the agent.

A reward is issued per successful interaction between certain 'actors'. The reward distribution is set based upon the type of contract that is issued. These transactions allow ABS to be distributed to multiple individuals and can in turn be sold. This generates volume, formulating a new cycle.

Coin Specification

	Mainnet	POV Network
Ticker	ABS	-
Algorithm	Lyra2REv2	-
Block Time	120 Seconds	60 Seconds
TPS	70	320
Block Size	2MB	4MB
Difficulty Algorithm	Dark Gravity Wave 3.0	-
Masternode Collateral	-1000* (V12.2.2a) -2500* (V12.2.3a)	-5,000 ABS -10,000 ABS
Governance Fee	5 ABS	-
Governance Minimum Quorum	10% of Masternodes	-

*The current version V12.2.2a is using a Masternode collateral of 1000 ABS. This will be replaced by a 2500 ABS minimum requirement in the next version (V12.2.3a) which will then become the new base tier.

Network Interaction

Although the POV network is totally transparent, there is a collateral requirement that must be met to allow access to the POV Masternode network. These nodes process the POV transactions and administer contract escrow. They also provide agents, enablers, users and the network with rewards based on the interactions.

In return for processing a POV interaction, the network will receive an additional reward on top of the block reward. In times of high network transactions, this will mean an increased ROI for POV Masternodes.

The POV network controls the escrow funds held for each contract. When a contract interaction occurs, the POV network commands the POV nodes to release escrow funds to the required interactors at specific periods during the day.

All transactions on the POV network are used to process ABS in an efficient way. This means that when there is high activity on the POV network it will have minimum effect on standard transactions on the ABS Mainnet.

The POV network is controlled and maintained directly by the POV Masternodes including block sealing. There is no reward structure on this network, its primary use is to facilitate payment on the Mainnet and reward 'actors'.

Network Terminology

The POV network is made up of a number of ‘actors’. These are people that interact with the network to successfully process a transaction.

- **Agent**

Agents are the contract initiators. They purchase ABS from an exchange and then issue contracts based on their needs.

- **Enablers**

Certain contracts require ‘enablers’ to create adverts and distribute media, either in a physical form or online. If they have met the requirements of the contract and a lead is generated, they will receive a reward for their interactions. These rewards depend on the type of contract they are fulfilling.

- **Users**

These are end users who communicate with the enablers or agent via the network to produce an interaction. It is possible for users to be rewarded based on their interaction with certain contracts.

- **POV Nodes**

A POV node is the core network processor. Masternodes receive rewards for processing interaction information on the network.

Tier network reward structure

To keep the network fair for all users, there will be equal share of the block reward based on a 1:1 ABS distribution offset against collateral. The upper tier nodes will receive rewards from the POV network as well, which will increase the ROI over the base tier.

Base Tier Node collateral

- 2500 ABS

Upper Tier Node collaterals (POV Nodes)

- 5,000 ABS
- 10,000 ABS
- 20,000 ABS (To be added at a later date)
- 40,000 ABS (To be added at a later date)

Proof of View (POV)

Proof of View allows network agents to form advertising, promotion or event based contracts. Agents must have an active connection to the network either via the wallet or via a website that is connected to a POV node. This website can be set up by Masternode holders that have the highest tier nodes. Contracts can be formed based on the following areas:

- (LOC) Location based contracts qualified via an Apple or Android wallet
 - These rewards are based on an agent trying to attract people to a location for a specific reason such as, but not limited to, an event, conference, party or festival.
- (PHY/ PHY-LOC) Physical based capture contracts qualified via an Apple or Android wallet. These are split into two areas based on pre-defined interactions:
 - (PHY) Media capture via a smart phone with the wallet installed. Before the campaign starts, adverts are catalogued and imported into the POV system, receiving an individual ID. The wallet will then match to this ID and will supply links to the advert which will generate a lead for the agent.
 - (PHY-LOC) Location media capture via a smart phone with the wallet installed. These must be in a certain location to receive rewards. This is a double verification process which combines standard physical based capture (PHY) and location.
- (SAW) Social media and website based lead generation contracts. These come under two different criteria:
 - (SAW-U) Unqualified Leads – these interactions are not subsidised via the collateral buffer and can only be funded via the agent's collateral alone:
 - Website adverts and banners processed via the network when an interaction occurs.
 - Outbound media, based on click per view.
 - (SAW-Q) Qualified Leads - these interactions are subsidised via the collateral buffer and can only be funded via the agent's collateral:
 - Lead Generation which results in a detailed form submission or website order attached to the contract ID.
- (UES) Users can become enablers by republishing media if the contract allows it.

Most contracts are subsidised by the block reward structure via the collateral buffer.

Collateral Buffer

The Collateral buffer is a stabilisation fund that bridges between the block reward and the POV network. It provides two outputs:

- A Proof of View contracts subsidy. This allows a subsidy percentage based on the level of engagement in the POV system. High demand will reduce the subsidy to a lower percentage. Lower demand will increase the percentage until 20% of the contract value is met.
- The second is an average output to counteract the dynamic adjustment algorithm that keeps the buffer full. This makes it fairer for Masternodes that received funds from the buffer overflow to receive a stable reward figure daily. This amount is split between the total number of nodes that have been active for more than one day and is transferred to all nodes on a daily basis.

Loading conditions

The reward structure is dynamically adjusted to allow the Collateral buffer to stay full based on demand. Here are a few loading condition examples:

Block Reward Allocation						
POV Loading	MN	Collateral Overflow	Collateral Funding	POW	POS	Governance
None*	40%	28%	2%	10%	10%	10%
Medium	40%	15%	15%	10%	10%	10%
High	40%	0%	30%	10%	10%	10%

POV Loading Conditions explanation

- None – The buffer is full and awaiting a smart contract to be initiated. All additional supply from the 28% allocation is diverted to the Masternodes and paid daily to all qualifying nodes.
- Medium – The buffer is in use but requires funding from the reward structure to stay at the algorithm's specific target. As only 50% of the block funding is required, the other 50% is dispensed to the Masternodes and paid once a day to all qualifying nodes.
- High – The buffer is being depleted rapidly and all available funding is initialised to restore collateral within the buffer.

The buffer will always pull a minimum of 2% from the block reward. This allows growth of the fund over time in a sustainable way. The base size of the fund is allocated via Governance. The subsidy algorithm tracks demand for the fund and in turn increases or decreases the size of the fund and subsidy percentage accordingly.

POV Reward

The POV reward system is funded by agents that wish to create smart contracts. Enablers, POV Nodes and users then receive percentages of the POV reward structure based on the type of interaction that is occurring in the contract.

Subsidy amounts are adjusted based on loading conditions; the POV network will always receive a base percentage per contract.

Below is a standard overview of the reward structure for different contract areas based on full load.

Contract type	Subsidy (SB)	Enabler	User	POV Network	Governance
(LOC)	20% MAX	-	88%	10%	2%
(PHY)	20% MAX	53%-(SB%)	40%	(SB%)+5%*	2%
(PHY-LOC)	20% MAX	53%-(SB%)	40%	(SB%)+5%*	2%
(SAW-U)	-	93%-(SB%)	-	(SB%)+5%*	2%
(SAW-Q)	20% MAX	73%-(SB%)	20%	(SB%)+5%*	2%
(UES)	20% MAX	53%-(SB%)	40%	(SB%)+5%*	2%

*The POV network percentage will always be 5 % higher than that of the subsidy, except for LOC based contracts.

Reward Allocation based on Tier level

Tier Level	POV allocation from interaction per click	Governance Sway per ABS
5,000	30%	+2 %
10,000	70%	+5%

Reward scenario for POV

The table below is an average reward based on 10,000 ABS being spent in one day on the POV network. This is assuming that all contracts are equal in volume (5.8% POV Fee). This reward can be added to the standard block reward. These are extreme conditions based on each end of the loadings range.

Tier Level	ABS Contract Average from one Day			
	No Subsidy	Reward in ABS	Max subsidy	Reward in ABS
5,000	1.74%	174	20% + 1.74%	2174
10,000	4.06%	406	20% + 4.06%	2406

This is the total reward, which would need to be distributed equally across the tier level.

Sources and Links

Github

<https://github.com/absolute-community/absolute>

Official Website

<https://www.absolutecoin.net>

BCT Announcement thread

<https://bitcointalk.org/index.php?topic=4418859.0;topicseen>

Reddit

<https://www.reddit.com/user/AbsoluteCoin>

Discord server

<https://discord.gg/FhtgzY8>

Telegram

https://t.me/absolute_community

Contact

contact@absolutecoin.net

Legal

Cryptocurrency investments are inherently high risk. Please make sure you are aware of the nature, complexity and risks before using any cryptocurrency.

Do not invest more than you can afford to lose. It is important not to use coins without taking into account the possible loss, since the type of change in these currencies is highly volatile and the Absolute team is unable to regulate market-defined prices. We strongly suggest seeking advice from your own financial, investment, tax or legal adviser.

The Absolute Development Team will always act in good faith and is not liable for the use of Absolute by other community members, persons or institutions.

References

1. The Proof of View system - Created and published by CryptoCentric, part of the Absolute Coin (ABS) development team: cryptocentric@absolutecoin.net
2. Bitcoin Foundation - <https://bitcoin.org/en/developer-documentation>
3. Dash - <https://dashpay.atlassian.net/wiki/spaces/DOC/overview?mode=global>

Basic Attention Token (BAT)

Blockchain Based Digital Advertising

Brave Software

March 13, 2018

Abstract

Digital advertising is broken. The marketplace for online advertising, once dominated by advertisers, publishers and users, has become overrun by “middleman” ad exchanges, audience segmentation, complicated behavioral and cross-device user tracking, and opaque cross-party sharing through data management platforms. Users face unprecedented levels of malvertisements and privacy violations. Mobile advertising results in as much as \$23 per month in data charges on the average user’s data plan, slow page loads, and as much as 21% less battery life. In response, over 600 million mobile devices and desktops (globally) employ ad blocking software and this number is growing. Traditional publishers have lost approximately 66% of their revenue over the past decade, adjusted for inflation. Publishers face falling revenue, users feel increasingly violated, and advertisers’ ability to assess effectiveness is diminished. The solution is a decentralized, transparent digital ad exchange based on Blockchain. The first component is Brave, a fast, open source, privacy-focused browser that blocks third party ads and trackers, and builds in a ledger system that measures user attention to reward publishers accordingly. Brave will now introduce BAT (Basic Attention Token), a token for a decentralized ad exchange. It compensates the browser user for attention while protecting privacy. BAT connects advertisers, publishers, and users and is denominated by relevant user attention, while removing social and economic costs associated with existing ad networks, e.g., fraud, privacy violations, and malvertising. BAT is a payment system that rewards and protects the user while giving better conversion to advertisers and higher yield to publishers. We see BAT and associated technologies as a future part of web standards, solving the important problem of monetizing publisher content while protecting user privacy.

Contents

1	Value Proposition	3
2	Introduction	3
2.1	An Inefficient and Troubled Market	3
2.2	The Attention Marketplace:	4
3	A New Deal: Attention-based Economics on Blockchain	12
3.1	Basic Attention Metrics (BAM)	13
3.2	Token Technology	14
3.3	Tokens Used as Publisher Payment	16
3.4	Tokens for User Applications	17
3.5	Roadmap	18
4	Business landscape	18
4.1	Competition	18
4.2	BAT Advantage Matrix	19
4.3	BAT Overview	19
4.4	Key Team Members	21
5	Token Launch	22
5.1	Token Launch summary	22
5.2	Token Distribution	22
5.3	User Growth Pool	22
5.4	Budget Allocation	23
6	BAT FAQs	24
7	Appendix	26
7.1	A More Efficient Market: Coase Theorem	26
7.2	A Three-Way Coasean Bargain	29
7.3	An Analysis of the Stability of the BAT	32

1 Value Proposition

We propose the BAT as a token of exchange in a secure, anonymous, opt-in advertising system based in the browser and the mobile app webview. The BAT system provides:

- Users: strong privacy and security when viewing advertisements, improved relevance and performance, and a share of tokens.
- Publishers: improved revenue, better reporting, and less fraud.
- Advertisers: less expensive customer attention, less fraud, and better attribution.

2 Introduction

“Attention has been widely recognized as a commodity, like wheat, pork bellies or crude oil. Existing industries have long depended on it to drive sales. And the new industries of the twentieth century turned it into a form of currency they could mint. Beginning with radio, each new medium would attain its commercial viability through the resale of what attention it could capture in exchange for its ‘free’ content.” -Tim Wu, Attention Brokers

The promise of advertising technology (“ad-tech”) was to create a more efficient marketplace for attention. The hope was that the Internet, the latest kind of “new medium,” would arrive with a transparent and efficient ad marketplace.

In theory, excellence would be rewarded. The best journalism and entertainment would receive the attention and funding it deserved. Ad tech would “get marketers closer to their users via data analysis, immediate valuation and distribution.” Data would be used to “accurately identify audiences, determine the value of those audiences, and deliver the right messages to them instantly.” [1] In short, users’ attention would be valued properly.

That didn’t happen. Instead, the ad-tech ecosystem that has evolved over the last two decades is a bewildering variety of middlemen and complexity. Worse, ad-tech introduced a host of correlated problems for publishers, advertisers and users. Users have lost their privacy, face increasing malware, pay high charges to download ads, and suffer slow speeds. Publishers have lost billions in revenue while fraud has skyrocketed. And advertisers face poor reporting and targeting.

This paper will review the current state of ad-tech and the predicament of content producers. It will outline a new solution that creates a transparent and efficient Blockchain-based marketplace for publishers, advertisers and users, accurately valuing and rewarding the key driver of Internet content: durable user attention.

2.1 An Inefficient and Troubled Market

Thomas Davenport and JC Beck note that “attention is focused mental engagement on a particular item of information. Items come into our awareness, we attend to a

particular item, and then we decide whether to act.”[2] Attention is, in this sense, a form of scarcity, which raises fundamental economic questions, which we shall address momentarily.

Advertising, throughout history, has been used as the primary mechanism to capture Attention, raise it to a level of Interest to incite some Desire that can then translate it into Action – otherwise known as AIDA.[3] The earliest forms of advertising date to ancient China, Egypt and the Middle Ages in Europe. The print form of advertising began to expand widely with the growth of 19th Century printed products. This marketplace of advertisers, publishers and users remained relatively straightforward – despite some additions – even as the new media of radio and television arose.

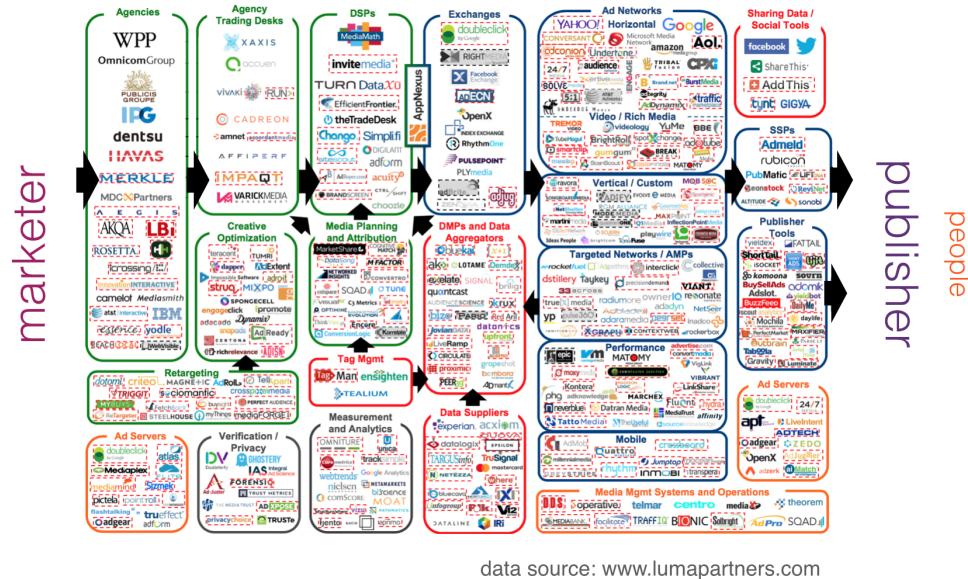
The rise of the Internet brought the development of a new level of advertising technology with the promise of higher speed and better information, two critical elements that had the potential to radically improve the efficiency of the attention marketplace. Somewhat counter-intuitively, the sheer complexity and opacity that organically developed has brought the opposite result. The system isn’t working as it should. As the Chief Brand officer of the largest advertiser, P&G, said recently:

“The days of giving digital a pass are over. It’s time to grow up. It’s time for action.”[4]

Especially in the last decade, the advertising ecosystem has become more complex and crowded, with many more players taking a piece of the advertising pie, either directly or indirectly. The complexity of this ecosystem increases the cost in headcount and difficulty of the tasks for the digital marketing teams on the advertiser’s side. At the other end of the system, the typical publisher faces both a shrinking market for the ad-blocker-free attention, and a shrinking slice of the advertising revenue pie due to the multitude of third party players who act as economic middlemen in the transaction.

2.2 The Attention Marketplace:

Sales planners currently budgeting for brand advertising are required to account for an excessive number of intermediaries that stand between the ad and the end user. Agencies, trading desks, demand side platforms, desktop and mobile network exchanges, yield optimization, rich media vendors and partnered services often consume significant portions of creative and delivery ad budget. It is also common for agencies in charge of packaging brand campaigns to use data aggregators, data management platforms, data suppliers, analytics, measurement and verification services to fight fraud, enhance targeting, and confirm attribution. These factors add up to a high transaction cost on the efficient provision of attention to brand ad campaigns.



data source: www.lumapartners.com

Publishers also face a number of costs and intermediaries on the receiving side of the ads served. Publishers pay ad serving fees, operational fees for campaign setup, deployment and monitoring, publisher analytics tools; also they give up substantial revenue to some of the same intermediaries that the brand advertisers use via programmatic ads. Publishers face direct costs of user complaints when malvertising spreads from exchanges to loyal readers, often with little or no idea of origin and with no help from the ad exchanges responsible for allowing such ads to serve from their systems. These diminish net revenue as the overall complexity of the advertising ecosystem raises headcount and expense.

There is a hidden cost to this complexity. A single ad unit may bounce across many networks, buy and sell-side ad servers, verification partners and data management platforms. Publishers lose revenue from each middleman transaction. Each one of these transactions also detracts from the user experience. Many of the middle players involve data transfers, which add latency. Any transfers done via script on page eat into the user's data plan and battery life on mobile. Users often find their experience further diminished when the results finally arrive, confounded by a bewildering array of distracting ads the publisher allowed to be placed in hope of greater revenue.

In addition, the violation of user privacy exacts a significant social cost; economists have compared violations of user privacy as analogous to environmental pollution.[5] According to Pew Research, “Fully 91% of adults agree or strongly agree that users have lost control of how personal information is collected and used by companies.” [6] A large majority, 64%, believe that the “government should do more to regulate advertisers” regarding how they use and store personal information. This is not surprising, given that a visit to a popular media site can often have 70 trackers set loose on the reader.

Fraud is also a major problem afflicting the advertising marketplace. Hackers create malicious bots that produce bogus websites that fool advertisers. Internet “bots,”

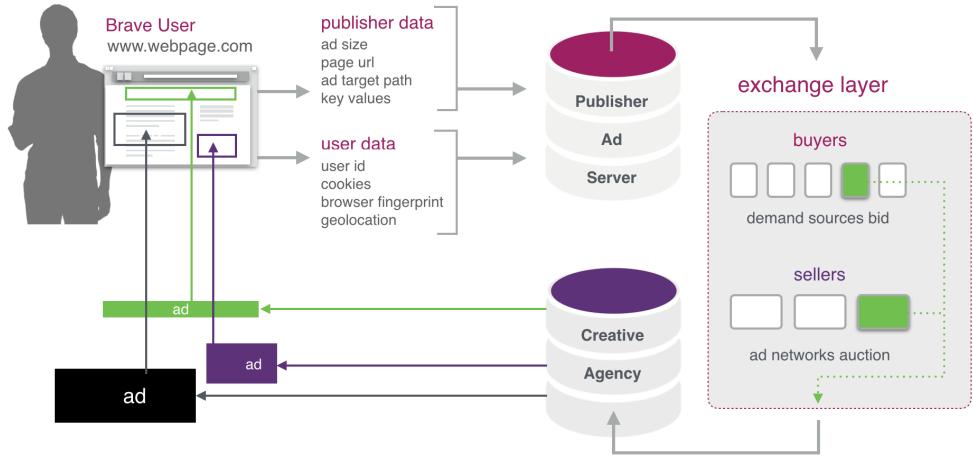


Figure 1: Typical Digital Ad Flow

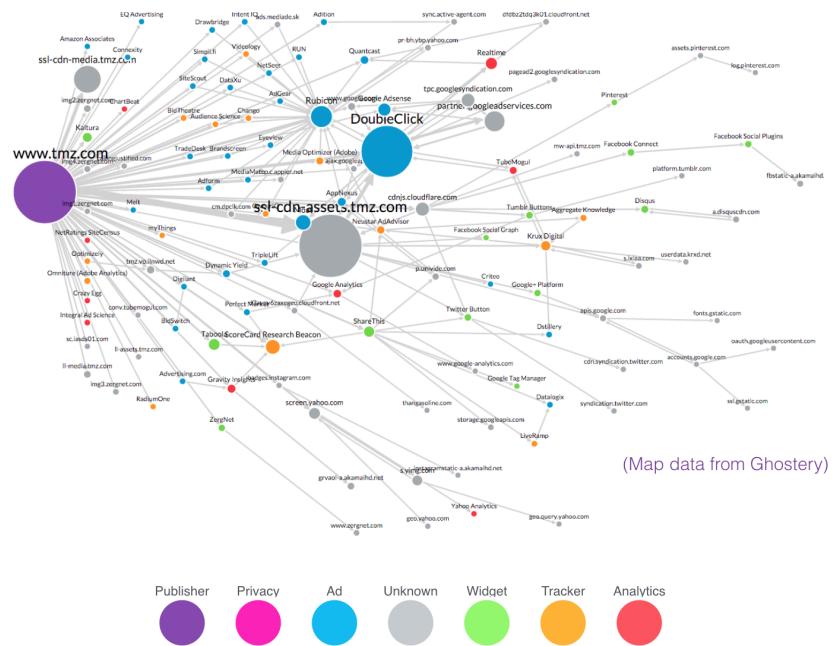
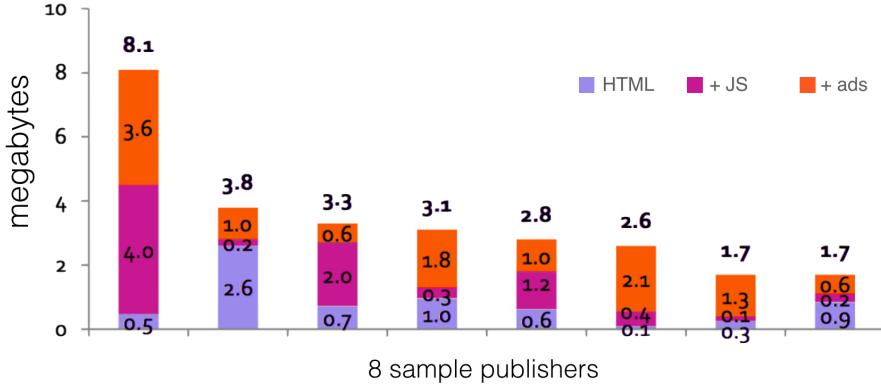


Figure 2: Typical Tracking on Large Content Sites



Note: Data was attributed by loading full pages, pages without ads or JS elements

data source: Enders Analysis

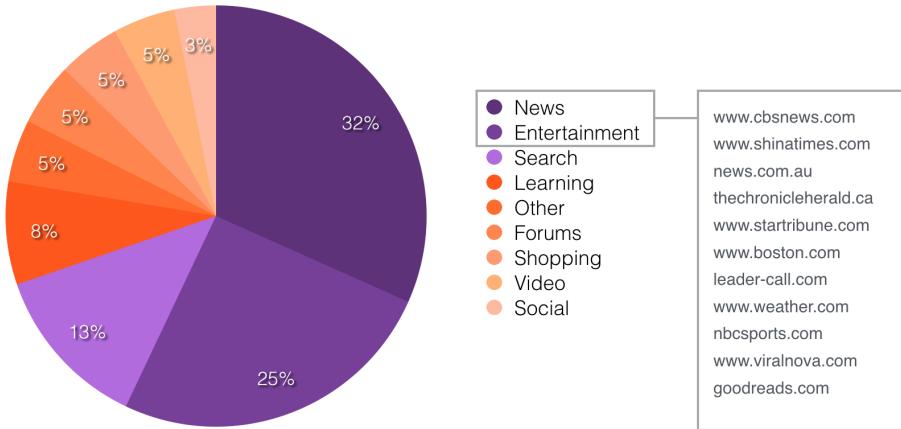
Figure 3: Data Transferred by Data Elements on News Sites

remote-controlled software running on compromised personal computers or cloud infrastructure programmed to engage in criminal activities -siphon billions of dollars each year from the ad industry. According to Business Intelligence: “These bots create websites filled with infringed content and generate fake traffic through a complex network of infected computers. In 2016, ad fraud created by internet bots is expected to cost advertisers \$7.2 billion, up from \$6.3 billion in 2015, according to a report from the Association of National Advertisers (ANA) and White Ops.”[7] There is no sign of this level of fraud leveling off or reducing.

Advertisers face fraud, while users are increasingly encountering malvertisements. Malvertisements are fake ads that trick users into clicking on them and then downloading malicious code, including ransomware. They can also entice users to visit fake domains used to steal financial information. According to a RiskIQ report released last year, “malvertising advert rates [rose] by 132% from 2015 to 2016.” The sites most frequently hit by malvertising, according to Bromium[8], are news and entertainment sites.

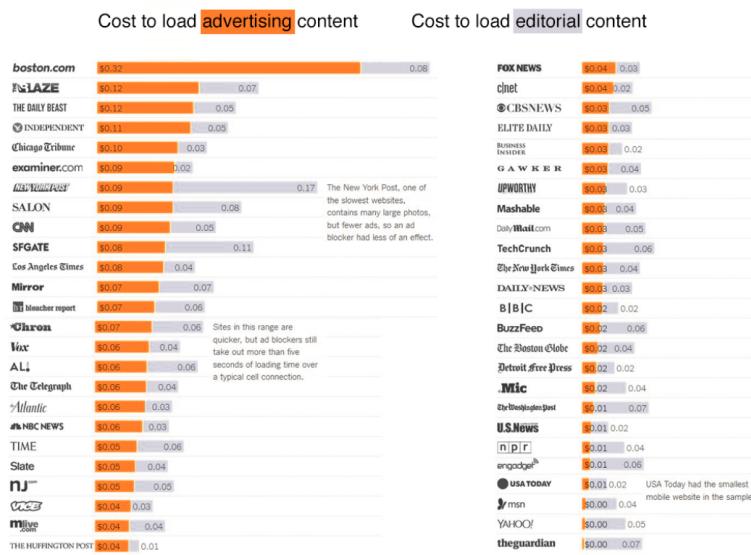
Web users are also not fully aware of the costs they pay for privilege of seeing advertisements. According to Business Intelligence, one study found that up to 79% of mobile data transferred during visits to popular publishers was a result of ads. The researchers compared data usage when a full page loaded without an ad blocker, with an ad blocker, and with an ad blocker and JavaScript disabled.

The article noted that the researchers concluded that “advertising accounts for half of all the data used by publisher pages loaded over mobile data networks” during the tests. The average smartphone user consumes 1.8GB a month. Based on carrier plans for 2Gb, this means that average users end up paying up to \$23 a month to download ads, trackers, scripts and other related data.[9]



data source: riskmanagementmonitor.com

Figure 4: Sites Most Frequently Hit By Malvertising



data source: New York Times

Figure 5: Content Loading Cost Comparison

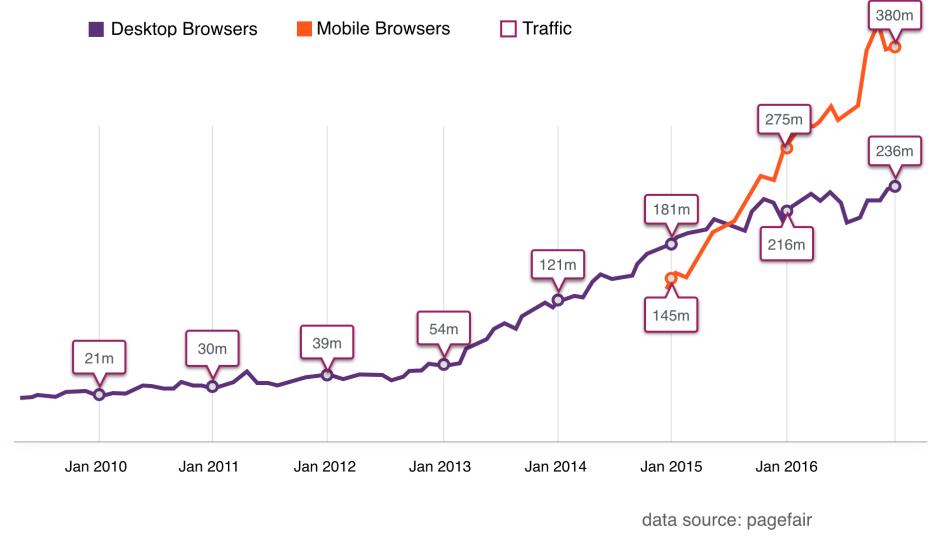


Figure 6: Ad Blocker Growth by Device

A study by the New York Times found the data used by advertising resulted in significant download times and costs across 50 top publishing sites. On one extreme, www.boston.com took 30.8 seconds for advertising and 8.2 seconds for editorial. The article concluded that removing ads saved “more than five seconds of loading time over a typical cell connection” for the articles studied. The data to load the ads came with a financial cost as well – the price for the advertising content often outweighs that of editorial material.

The sum total of malvertisements, load times, data costs, battery life, and privacy loss has driven users to adopt ad-blocking software. This further reduces publisher revenues and leaves the remaining ad-viewing audience even harder to target.

Ad blockers are a growing problem for publishers. Studies confirm that users of ad blocking software prefer the simplicity of navigation of ad-free or nearly ad-free content.

Over 600 million mobile and desktop devices now use ad blocking, according to Pagefair. It is projected that 86.6M Americans will use an ad blocker in 2017[10]. Younger users are also more likely to adopt ad blocking technology, making the long-term financial impact of this technology worse than it appears at first glance[11].

This “perfect storm” for publishers has only gotten worse over the last few years as Google and Facebook have taken more and more share of advertising revenues. Together they claim *over 73% of online digital ad revenue, and an astounding 99% of all growth from 2015 to 2016 in US total online ad budget*[12]. The increased attention for publishers brought by Google and Facebook would seem to be a net positive. But the traffic driven by social media is of lower quality than direct links. Users who arrive at a news site from social media typically only engage with the site for a third[13]



Figure 7: Demographics of Ad Blocker Usage

of the time compared to those who are direct visitors. Distributed content hosting makes up only 14% of publisher revenues, with the majority of the revenue coming from YouTube[14]; many publishers have experienced serious commodification problems with these platforms.

Advertisers on these platforms also face serious challenges. The sheer size of the platforms make them opaque and difficult to assess the effectiveness of advertising campaigns on their platforms. Since most of the analytics products targeting these platforms are provided by the platform owner, principal-agent conflicts arise. Some advertisers have decided that traffic coming from the walled gardens isn't worth the trouble. Some have even suggested based on third party analytics that a large proportion of the traffic is without value to the advertiser[15].

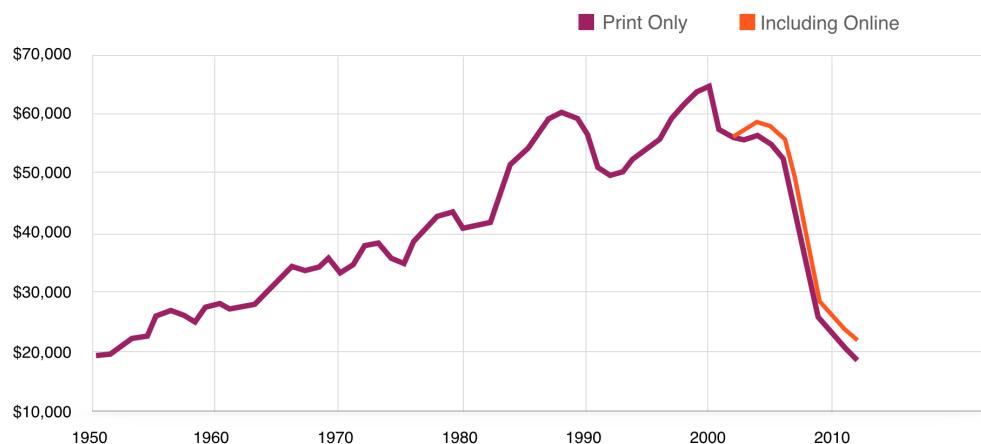
In an effort to expand their walled gardens and to reinforce market dominance by traffic and data otherwise ingested from users directly on the publisher domain, major platform players have begun offering alternative content delivery channels with claims of incentivized placement and a faster, more secure user experience. While Facebook Instant Articles, Google AMP project and Apple News delivery channels were initially presented to publishers as opportunities to extend reach and visibility, they ultimately diminish publishers' control of their brand narratives and reader relationships, and divert direct attention away from publisher sites over the long run.

Generally speaking, the publishing industry faces an existential threat. Legacy publishers have faced declining revenues for decades. Pressures on publishers to create content optimized for clicks has resulted in cut-backs to long form articles, investigative journalism, and foreign news bureaus, and has spawned the much lamented social cost revealingly named "clickbait." This dysfunctional dynamic has been noticed across the industry. Marketing budgets continue to climb[16], yet publisher revenues are static or shrinking[17]. This indicates serious market inefficiencies which can be repaired with a

U.S. Ad Revenues	Q3 2015	Q3 2016	Growth	Share of Growth
Google	\$7.9B	\$9.5B	\$1.6B	54%
Facebook	\$2.1B	\$3.4B	\$1.3B	45%
Everyone Else	\$4.6B	\$4.7B	\$40M	1%
PWC/IAB	\$14.7B	\$17.6B	\$2.9B	

data source: fortune.com

Figure 8: Ad Revenue for Google vs Facebook vs Others



data source: Newspaper Association of America

Figure 9: The Fall of Newspaper Ad Revenue

simplified and more efficient economic system based on new technologies.

3 A New Deal: Attention-based Economics on Blockchain

The diversity of middle-men and the lack of value-add to the publisher and user make some sort of simplification of the present online advertising ecosystem inevitable. Present trends are toward an oligopoly where gatekeeper companies such as Google and Facebook control the entire online marketing budget with publishers powerless to control their revenues. Also, as users continue to adopt ad blocking technology the consequent shrinking of the remaining ad-funded market seems inevitable.

The reality remains: user attention is valuable, but it hasn't been properly priced with an efficient and transparent market system. While it has become a platitude that vast amounts of information are generated on and by the Internet, human beings are only able to devote a limited amount of attention to certain small subsets of the information. Information in the modern age is relatively cheap. Human attention paid to the information is the rare quantity. As Herbert Simon put it in an influential 1971 article:

“...in an information-rich world, the wealth of information means a dearth of something else: a scarcity of whatever it is that information consumes. What information consumes is rather obvious: it consumes the attention of its recipients. Hence a wealth of information creates a poverty of attention and a need to allocate that attention efficiently among the overabundance of information sources that might consume it.”

Ultimately, a publisher provides information which may be of value to the user. Users give attention to the publisher in return for information that they value with their attention. At present, the publisher is paid by monetizing attention via a complex network of intermediary players through ad networks and other such tools. The publisher isn't paid directly for the attention given by the user. The publisher is actually paid for the indirectly measured attention given by users to ads. Publishers are used to working with this model for print ads, but web ads remain problematic for many of the reasons stated above. Users are subjected to the negative externalities that come with the present advertising ecosystem.

Users thus suffer a form of “electronic pollution” consisting of threats to security, threats to privacy, costs in inefficient download times, financial costs in extra mobile data fees, and in the case of the many ads, excessive costs to their attention. Human attention can be exhausted, until dopamine levels recover. Neurons can and do learn to ignore ad slots (so-called “banner blindness”). Abuse of user attention and permanent loss of users, via ad-slot blindness and ad-blocker adoption, make attention different from substitutable commodities such as pork bellies or crude oil, in the final analysis. While most users may be willing to pay some price for access to the publisher's information, user attention is mispriced when we sum up the growing negative externalities imposed by the present advertising ecosystem.

3.1 Basic Attention Metrics (BAM)

To improve the efficiency of digital advertising requires a new platform and unit of exchange. The first phase involves the roll-out of a new browser, Brave, a fast, open source, privacy-focused browser that blocks invasive ads and trackers, and contains a ledger system that anonymously measures user attention to accurately reward publishers. The next phase involves the introduction of Basic Attention Token or BAT. It is a token for the decentralized ad exchange. BAT connects advertisers, publishers, and users, creating a new, efficient marketplace. The token is based on Ethereum technology, an open source, blockchain-based distributed computing platform with smart contracts. These cryptographically secure smart contracts are stateful applications stored in the Ethereum blockchain, fully capable of enforcing performance. The token is derived from – or denominated by – user attention. Attention is really just focused mental engagement – on an advertisement, in this case.

The ability to privately monitor user intent at the browser level allows for the development of rich metrics for user attention. For example, it is known whether an impression has been served to an active tab, and measure the seconds of active user engagement. Attention is measured as viewed for content and ads only in the browser’s active tab in real time. The Attention Value for the ad will be calculated based on incremental duration and pixels in view in proportion to relevant content, prior to any direct engagement with the ad. We will define further anonymous cost-per-action models as the system develops.

In-device machine learning will match truly relevant ads to content from a level that middlemen with cookies and third party tracking are unable to achieve, regardless of how much of the user data is extracted and monitored from external models. These external models are still unable to track transactions well enough not to serve ads for products users have often already purchased. User engagement through genuine feedback mechanisms ensures that users that have opted in for BAT are getting the best possible product match that they’re most likely to convert into a transaction. Ultimately it comes down to trust and respect with and for the user. By keeping the data on the device only, encrypting the data and shielding the identities of our users as a core principle, BAT forms a bond with users that proves that not only does their data hold value, it holds substantial value that has been ignored and exploited by the middlemen year after year in the current industry model.

Several scoring algorithms have been tried with the Brave donation ledger system, which automatically donates an amount proportional to the attention given to a website.

One of the metrics suggested is 5 total views of advertising content in an active window, for at least 5 seconds each. Hits of this nature would be calculated on a 30-day moving window.

Another suggested metric is the “concave” score[18]. This is a score which rewards a publisher for a thresholded and bounded function of the amount of time spent with the open and active page. For example, one “point” could be awarded for a two second view of the page, with two points for a 30 second view, and 3 for a 60 second view, with diminishing or bounded returns for longer views.

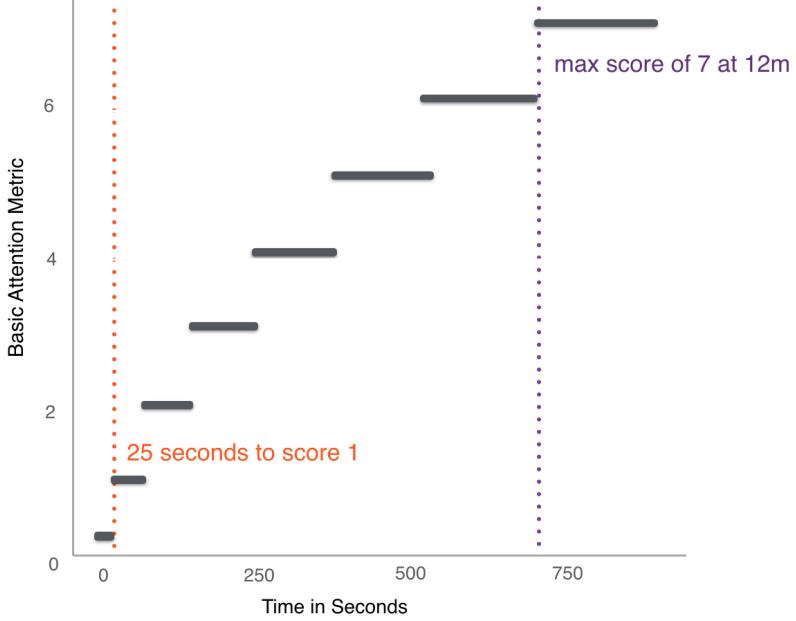


Figure 10: Basic Attention Metric Score Over Time

The present implementation of the concave score, which is being used to distribute attention metered donations to the publishers, is a thresholded, time limited quadratic score. The formula is as follows:

$$score = \frac{-b + \sqrt{b^2 + 4a * duration}}{2a}$$

where $a = 13000$, $b = 11000$ and $duration$ is measured in milliseconds. This gives a minimum threshold of 25 seconds to achieve a score of 1. The upper bound is set to be around 12 minutes of attention given to the article, with a maximum score for a given piece of content of 7. This can be seen in figure 10.

Another potential metric is a targeted ad based on a subset of keywords purchased at the advertising partner end, combined with the attention metric, essentially selling the attention along with an advertising topic.

We expect publishers and advertisers to suggest new metrics of user attention to be surfaced, and encourage other vendors to build on the topic as we progress.

3.2 Token Technology

The Basic Attention Token (BAT), a token based on Ethereum, is an important element of a new marketplace. Ethereum is an open source, blockchain-based, distributed computing platform oriented towards smart contracts. Effectively, Ethereum is a dis-

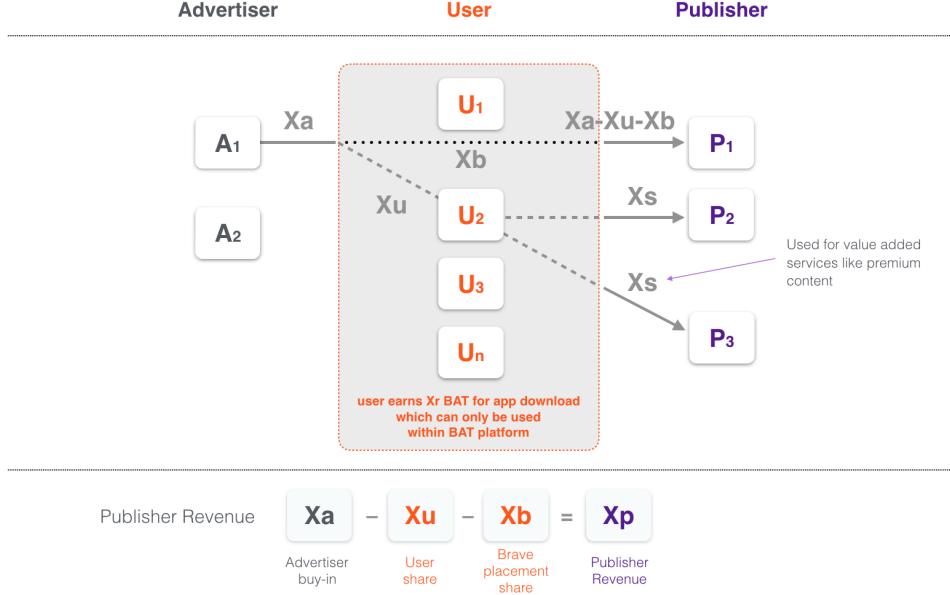


Figure 11: Value Flow of the Basic Attention Token

tributed virtual machine that allows end users to construct smart contracts for transactions. Smart contracts are stateful applications stored in the Ethereum blockchain. These contracts are cryptographically secure and can verify or enforce performance of the contract. Token contracts are a standard feature of the Ethereum ecosystem.

Ethereum has been used for mobile payment systems, distributed exchanges, tokens pegged to commodities and fiat currencies, market clearing mechanisms, micropayment systems for distributed computing resources, commodities and securities exchanges, crowdfunding, and legal document verification. Large firms have invested in and deployed Ethereum, with JP Morgan, Deloitte, IBM, Santander Bank, Microsoft, the Luxembourg Stock Exchange, and the Royal Bank of Scotland being key early adopters.

Micropayments using BAT will be accomplished for the first stage deployment with the Brave Micropayments Ledger. Each viewed ad will be verified at the browser using the BAM.

This flow shows the conceptual flow of the BAT payments. The flow of the BAT payments will not follow this chart precisely in first iterations of the BAT payment system as the payments will be regulated by the Brave ledger system, but the total effect will be the same. The high-level concept is the advertiser sends a payment in token along with ads to users in a locked state X_a . As the users view the ads, the flow of payments unlocks, keeping part of the payment for their own wallet (X_u), and passing on shares of the payment to Brave (X_b) and passing the remainder on to the Publisher (X_a-Xu-X_b).

The BAT will, in early stages, be specifically tied to Brave browsers and Brave

servers, along with verified publishers. Ad fraud will be prevented or reduced by publication of source code and cryptographically secure transactions. Ads served to individual browser/users will also be rate-limited and tied to active windows and tabs. Payments in BAT will be sent only to publishers, though a payment for viewing an ad on one publisher may be used at another publisher or kept for some other premium services supplied through the BAT system.

3.3 Tokens Used as Publisher Payment

Publisher payment will be through the BAT system. For the first deployment of BAT, the transactions in BAT will take place through the Brave Ledger system, which is an open source Zero Knowledge Proof scheme presently deployed to allow Brave users to make anonymous donations to publishers using bitcoin as the medium of exchange. The Brave Ledger system uses the ANONIZE[19] algorithm to protect user privacy.

For the first incarnation of BAT, all payments in BAT must have a publisher endpoint. The publisher client as it is coded today already measures user attention as described above. The “concave” awarding mechanism calculates an attention score based on a fixed threshold value for opening and viewing the page for a minimum of 25 seconds, and a bounded score for the amount of time spent on the page. A synopsis of user behavior is then sent back to the Brave Ledger System for recording and payments made on the basis of the scores.

Much of the infrastructure required to deploy BAT at the back end is presently code complete, in place and being used to distribute donations based on user attention. As such, this infrastructure will be leveraged to deploy BAT as soon as possible for testing, user, and advertiser feedback.

A fully distributed ledger is desirable, both for public accountability and potential scalability reasons. Publishers, advertisers and users of the BAT token will have incentive to use such a system to keep track of payments within the BAT system.

State channels allow for multiple small transactions with strong anonymity guarantees when using the correct matching algorithms. While Raiden and other state channel schemes are becoming integrated with the Ethereum ecosystem, and new blockchains such as Zcash and Monero offer stronger privacy guarantees with rapidly increasing feature sets, it is likely that a new scheme addressing the unique problems of this type of transaction will be used for large scale multiparty transfer of BAT.

A lottery system may be used, where small payments are made probabilistically, with payments happening essentially in the same way that coin mining works with proof of attention instead of proof of work[20, 21], BOLT[22], Zero Knowledge SNARK[23] or STARK[24] algorithms may become part of this stack for guarding privacy of participants. The BAT situation is mitigated by the fact that the privacy of the browser customer is of primary importance; publishers and advertisers have fewer privacy concerns. The transactions in a fully distributed BAT system will almost always be one to many and many to one, therefore novel zero-knowledge transactions may be suggested by this arrangement.

As Brave moves to a fully distributed micropayment system, we expect other devel-

opers to use our free and open source infrastructure to develop their own use cases for BAT. We want BAT and the tools associated with it to eventually become important web standards for future development of web content. Publishers, advertisers and users who view web content deserve a private, secure and well-engineered future.

3.4 Tokens for User Applications

As users are given access to some of the advertising spend in BAT, they will become an important and active part of the advertising and publishing economy, rather than the passive participants they are presently treated as. While tokens can be donated to individual content providers and publishers, there are any number of use cases for the tokens.

An obvious use case is for very specific targeted advertising. Many small businesses have modest requirements which may be well served by tokens they acquire through their normal browsing activities. Users may also find new uses with low barrier to entry highly targeted ads; personal ads targeting people of a religion or subculture for example.

Some publishers may have premium content they would ordinarily only offer to subscribers. Since subscription models are not typically favored by users on the internet, this could unlock new revenue for premium content providers. Content may also be bought for friends using the token; if someone likes a premium article, they can make a micropayment to send it to three of their friends.

Higher quality content may also be offered to users for a BAT transaction. For example, higher quality video or audio on an entertainment channel, or some kind of summary of headlines in a news source. Video or audio content in a news or other information source may be restricted to people who pay a small micropayment.

Comments may be ranked or voted on using BAT tokens, similar to the “thumb-up/thumbdown” on some comment sections. Comment votes backed by BAT may be given more credibility due to the fact that someone cared enough to back the comment with what would be a limited supply of token, as well as the fact that a token transfer can be verified as coming from real people rather than robots. The right to post comments may also be purchased for some minimal payment, to cut down on abusive commenters.

Eventually, BAT may be used within the Brave ecosystem to purchase digital goods such as high resolution photos, data services, or publisher applications which are only needed on a one-time basis. Many publishers have access to interesting data sets and tools which they are not able to monetize on a subscription basis, but which individuals may wish to occasionally use. For example, firms such as Pro Publica, Citizen Audit and Gartner contain interesting public data and premium content, but many individuals find a subscription too costly. Small parts of news archives may also be of interest to people who do not want to purchase access or a subscription to the entire archive.

BAT may also be used in games provided by publishers within the Brave ecosystem. While such applications are not presently popular with publishers, many platform providers have hosted profitable gaming applications. It could create a new economy of

app creators to go along with content. For example, 'punch the political/entertainment figure' games to go along with critical articles. People won't get out their credit card to use such an application, but they may be willing to part with some value they acquired in normal browsing activities to enjoy punching their favorite entertainment figure.

Custom news alerts may be offered as a service by news providers for a small payment of BAT within the ecosystem. Such news alerts may be very valuable to individuals who are concerned with current events, financial news or some anticipated event.

3.5 Roadmap

- Pre 1.0 BAT: Brave already has an anonymized ledger system for making donations and payments to publishers based on user attention. The secure vault using the ANONIZE algorithm to ensure customer privacy is an important piece of the BAT ecosystem which is already in place and deployed in Brave. Brave is already measuring user attention at the browser level and distributing donations to the publishers using this system.
- 1.0 BAT: BAT wallet integrated with the Brave browser. Verification and transactions to be handled by Brave's internal Zero Knowledge Proof ledger system to protect individual user anonymity from advertisers, publishers and third parties. Ad inventory will be valued, and transactions will be calculated from reported Basic Attention Metric (BAM) data.
- Beyond 1.0 BAT: Make the transfer and verification process entirely distributed on Ethereum using a state channel scheme with Zero Knowledge Proof protocol for ensuring user privacy. Add alternate BAM metrics based on advertiser feedback. This will allow for full user privacy as well as a decentralized audit trail for advertisers, users and publishers to ensure they received correct payments for the advertising delivered through the BAT network.
- Browser as platform/BAT: Further BAM metrics based on advertiser feedback as needed. Partners building applications on the BAT infrastructure. Also, at this point we plan to explore value-added services that can be offered to users on the browser platform through BAT.

4 Business landscape

4.1 Competition

- Reddit Gold is a premium membership program, granting access to extra features to improve experience. Reddit is a major publisher, but this program is designed by and limited to Reddit. It does not offer publishers a mechanism for publishers and users to monetize through the use of Blockchain-based token.
- Steem is a social-media and blogging platform lets users earn revenue when they receive upvotes. It is a kind of monetized Reddit. Steem does use Blockchain, but

it is not a generalized means for publishers and users to be rewarded for content. In short, it is not a Blockchain-based digital ad platform. It is specific to the Steem platform.

- Blendle is a kind of iTunes for journalism, offering micropayments on a per-story basis. It gives readers a collection of stories based on preferences. Brave and BAT do not curate anything. Users merely go about their business on the web and publishers are rewarded. Blendle is not a token-based digital advertising platform.
- Google is a search engine company that makes most of its revenue from digital advertising. Google is at the center of the existing digital advertising ecosystem. They benefit from the complexity and opaqueness that defines it. BAT intends to empower the very users and publishers that are receiving less than they should. Google does not have a Blockchain-based tokenized system of offering rewards. Users are often unaware of how their privacy is compromised using Google.

4.2 BAT Advantage Matrix

Present ecosystem	BAT token ad payments
User frustration over loading time	Fast loads
Walled gardens	Free software, open source infrastructure
Bandwidth wasted	Low bandwidth overhead
Screen clutter	Uncluttered screen
Irrelevant ads	Ads tuned to user interests
Security issues	No malware
Viewability problems/attribution	Secure attribution/attention score
Advertiser uncertainty about delivery	Perfect delivery certainty
CPM/click based	Attention-based
Reader attention not valued	Reader is paid for attention
Publisher revenues lowering	Larger publisher revenues
Expensive ad buys due to middlemen	Efficient ad buys
Complex/expensive viewability metrics	Simple/free viewability metric
User's privacy violated	Perfect user privacy

4.3 BAT Overview

The Basic Attention Token (BAT) was developed to address the broken digital advertising market. BAT, an ERC20 token built on top of Ethereum, will be the unit of exchange in a new, decentralized, open source and efficient blockchain-based digital advertising platform. In the ecosystem, advertisers will give publishers BATs based on the measured attention of users. Users will also receive some BATs for participating. They can donate them back to publishers or use them on the platform. This transparent system keeps user data private while delivering fewer but more relevant ads. Publishers experience less fraud while increasing their percentage of rewards. And advertisers get

better reporting and performance. The first part of the solution, the Brave browser, is already operational. Brave is a fast, open source, privacy-focused browser that blocks invasive ads and trackers, and contains a ledger system that anonymously measures user attention aggregate to accurately reward publishers. The next step is introducing BAT.

Currently, we plan to utilize the Brave Browser for BAT, but other developers are free to utilize other browsers.

Brave is more than a browser: it defends your data on your devices and synchronizes your personal and private browsing profile across devices using client-side encryption. Your data, studied and abstracted by on-device-only machine learning, provides you with private and anonymous options to get compensated for your attention. Brave cuts out all third-party trackers and middle-players, eliminating data leakage, malware risk, and excessive fee-taking. Brave does this while providing publishers with a substantially larger revenue share than they are receiving in existing inefficient and opaque marketplaces.

Brave thus aims to reset the online ad-based Web ecosystem, giving advertisers, publishers and customers a win-win solution whose components and protocols can become future Web standards.

4.4 Key Team Members

- Brendan Eich, CEO, co-founded Brave. Created JavaScript. Co-founded Mozilla & Firefox.
- Brian Bondy, Lead Developer, co-founded Brave. Previously: Khan Academy, Mozilla, Evernote.
- Scott Locklin, Senior Engineer, Co-founded Kerf Software. Machine Learning, Forecasting & Quantitative Finance.
- Bradley Richter, Head of Design, Previously: EFI/Fiery, Co-creator: eBeam & Luidia, Percipo. Advising Circullio.
- Catherine Corre, Head of Communications, Previously: AOL, Netscape.
- Marshall T. Rose, Senior Engineer, PhD from UC Irvine, co-creator of SNMP and was with the Internet Engineering Task Force.
- Brian Johnson, Senior Engineer, was previously at JD Power and Korrelate.
- Luke Mulks, Senior Ad-tech Specialist, for technical incident response, investigation, support & issue resolution for ad tech and the Brave Browser. Developing/advising on ad tech and tracking threats that Brave shields users from (pr/blog).
- Aubrey Keus, Senior Engineer, Previously: Pulse360.
- Yan Zhu, Senior Engineer, EFF Fellow. Previously: Yahoo, Tor Project, HTTPS Everywhere, Privacy Badger.

5 Token Launch

5.1 Token Launch summary

Our goal is to raise a maximum of \$24 million USD and a minimum of \$5 million USD. Some of the numbers may change with ETH/USD exchange rates and volatility, but the following numbers are best effort estimates as of May 28, 2017.

- **Maximum financing:** 156,250 ETH -this may change with exchange rates.
- **Minimum financing:** 27,343.8 ETH.
- **Exchange rate:** 1 ETH = 6,400 Basic Attention Tokens (BAT) -this may change with ETH exchange rates.
- **Token contract address:** TBD (Published through various channels 48hrs before crowdsale launch date).
- **Launch date and time:** 8AM PST May 31, 2017 block number 3,798,640
- **Token launch time-frame:** 30 days (based on Blocknumber 3,963,480).
- **Token launch completion:** Token launch will end when either the maximum number of ETH are raised or block number 3,963,480 is reached. If less than the minimum ETH are raised, ETH can be retrieved by holders of BAT.

5.2 Token Distribution

- Brave: 13.3% of max; 200 million BAT.
- User growth pool: 300 million BAT.
- Token available to public at launch: 1 billion (corresponding to the ETH raised at token launch).

5.3 User Growth Pool

User growth fund is used to incentivize users to participate in the BAT ecosystem.

- A 300 million endowment is for early adopters of Brave and the BAT at up to 5 BAT/user.
- BAT received as a reward can only be used within the BAT ecosystem for value added services.
- Unused BAT after 6 months will be sent back to the user growth fund which can then be used for new users.
- Existing Brave users can get tokens by updating their app and verifying phone number.
- No new tokens will be created once the user growth pool is exhausted.

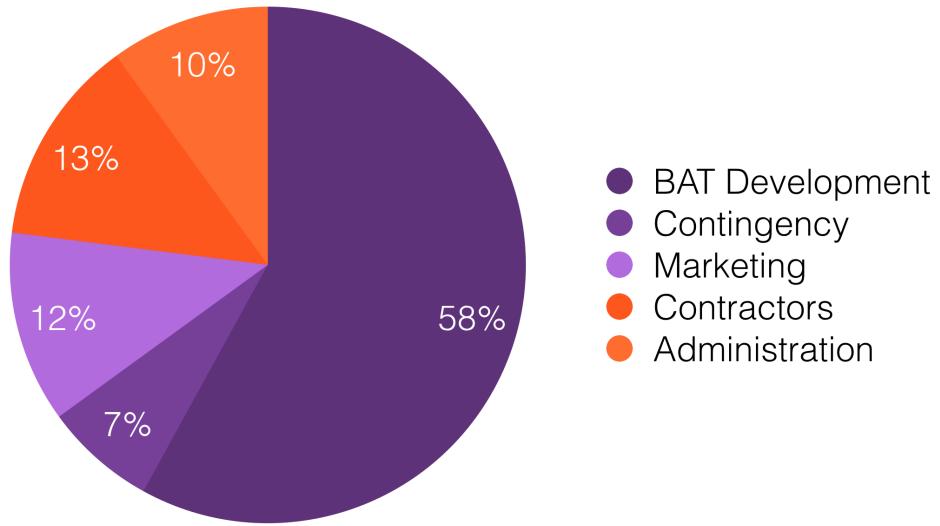


Figure 12: Budget allocation of ETH raised

5.4 Budget Allocation

- **BAT Team: 58% of budget** The team consists of just over 20 engineers. This financing allows for the rollout of the BAT solution, including the necessary adjustments to and development of the existing Brave browser technology.
- **Administration: 10% of budget** Consists of BAT legal, security, accounting and other associated administration costs.
- **Marketing: 12% of budget** Marketing will focus on expanding awareness and adoption of the Brave browser and the BAT solution among users, publishers and advertisers. This also includes the growth and maintenance of the world-wide community.
- **Contractors: 13% of budget** These funds will be directed at third-party providers offering engineering, marketing, growth-hacking, PR, partnerships, affiliate programs and more.
- **Contingency: 7% of budget** This is a set-aside for unforeseen costs.

6 BAT FAQs

What does BAT stand for and what is it?

Basic Attention Token. The BAT, a token based on the Ethereum technology, is a unit of exchange in a new Blockchain based digital advertising system. User attention is anonymously monitored in the Brave browser and publishers are rewarded accordingly with BATs. Users also get a share of BATs for participating.

What do BATs represent?

BATs are tokens in a new Blockchain and attention-based digital advertising platform. They are not refundable, nor are they securities or for speculation. There is no promise of future performance. There is no suggestion or promise that BAT has or will hold a particular value. BATs give no rights in the company and do not represent participation in the company. BATs are sold as a functional good. Any value received by company may be spent without conditions. BATs are meant only for experts in cryptographic tokens and blockchain-based software systems.

What amount is being raised? Whats the cap of tokens? Will there be a follow-on offering?

We are targeting a raise of as much as \$24 million USD and a cap of 1.5 billion tokens. We do not plan to have a follow-on offering.

What crypto-currencies are accepted in the crowdsale?

ETH will be accepted in the crowdsale. You will be required to have an Ethereum wallet pointed at the token/crowdsale address to participate in the crowdsale. BAT are Ethereum derived tokens. If you hold BTC or some other crypto-currency it can be exchanged for ETH and used to participate in the crowdsale.

When will the Crowdsale happen?

We're working with security auditors to finalize the contract. When they have completed their analysis we will announce the date. Note that the BAT crowdsale parameters will be tied to blocknumber, so times will depend on Ethereum mining rates. The contract will be pushed to Ethereum mainnet 3 days before the crowdsale starts. We'll also give people a week to interact with the contract on Ropsten/testnet.

What is the price of BAT?

BAT will be a fixed ratio to ETH. This may vary slightly with ETH volatility as we get closer to the contract deployment date. The exchange rate will be 6400 BAT per ETH.

How will Brave use ETH raised during token launch?

The ETH received in the crowdsale will be used by Brave Software to build out the Blockchain-based digital advertising system, which uses BATs as a unit of exchange.

How will Brave store ETH?

Brave will use the standard Ethereum multisig wallet to store ETH.

Are BAT tokens transferable?

Crowdsale tokens are immediately transferable. Tokens used in the Browser may only be donated or used to pay publishers for premium content or for other services. Tokens may also be used by publishers for promotions.

7 Appendix

7.1 A More Efficient Market: Coase Theorem

Problems involving social and transactions costs have been studied by economists. Ronald H. Coase was awarded the Nobel Prize in Economics in 1991 for his work on the allocation of radio frequency resources.[25] Modern problems in ad-tech are addressable using the work of Coase and subsequent commenters on his idea. At present, the effects of today's overcomplicated advertising ecosystem is a negative externality or "social cost" for the user. The user's privacy is invaded, the browsing experience compromised, and even the limited supply of internet bandwidth on mobile devices is depleted by the present state of this ecosystem. Effectively, the market for user attention has become inefficient; the transaction costs of advertisers purchasing attention have become too high.

The widespread adoption of ad blocking technology adds a negative externality on the publishers as well. If everyone blocked advertisements, there would be little content left to exchange for user attention, as publishers go out of business. An efficient market for attention would remove these negative externalities, or compensate all parties to the transaction in an efficient way.

The Coase theorem states that trade in an externality or "social cost" is possible. If there are sufficiently low transaction costs, information symmetry, and well defined property rights, bargaining will lead to a Pareto-efficient outcome regardless of the initial allocation of property.

The standard textbook example of the Coase theorem consists of a factory which produces pollution as a side-effect of the manufacturing process, and a neighboring landowner who suffers from the pollution.

In the case where the neighbor owns the pollution rights;

$$Q = 1 - (P + c)$$

c is marginal cost of production, P is price for pollution permit, Q is marginal cost function for the manufacturer in the case. Neighbor has valuation ν for clean environment, and the sale of Q pollution permits entails a loss of $\nu Q = \nu(1 - (P + c))$, so the neighbor finds the price of pollution permits by maximizing net benefit

$$\max_P \{(1 - (P + c))P - \nu(1 - (P + c))\}$$

The benefit maximization is

$$1 - 2P - c + \nu = 0$$

Giving the price

$$P = \frac{1 - c + \nu}{2}$$

and the units bought by the factory

$$Q = \frac{1 - c - \nu}{2}$$

If the factory has the entire property right, the neighbor effectively purchases some share of the pollution right from the factory which it doesn't use. The neighbor wants to buy $Q = 1 - (P - \nu)$ units. The factory maximizes its net benefit with

$$\max_P \{(1 - (P - \nu))P - c(1 - (P - \nu))\}$$

The factory's profit maximization is

$$1 - 2P + \nu - c = 0$$

So the price is still

$$P = \frac{1 - c + \nu}{2}$$

For Coase's theorem to hold symmetrically, it requires well-defined property rights. By definition, the user's attention is the valued quantity. The user can make the decision to block ads from a given publisher, or choose to forgo interacting with a publisher altogether.

This makes it obvious that attention belongs to users de facto and notwithstanding the efforts of some publishers and advertising firms to assert ownership of user attention de jure. Even in commonplace situations where user attention is de jure required, de facto, users still own their own attention. For example, attention is required while the safety demonstration is given on an airline flight, but people often ignore it anyway.

Another requirement for validity of the symmetric version of Coase's theorem is information symmetry. Information asymmetry between publishers, advertisers and users has kept the existing advertising ecosystem in place for some time, but as we can see from the growing use of ad-blockers, the information asymmetries on the user side are crumbling.

At present, advertisers and publishers have a severe information asymmetry in that most of the metrics they use to assess campaign effectiveness are indirect and administered by middlemen whose interests are not aligned with the interests of one or both parties. Complex "viewability" metrics create unnecessary conflict between advertisers and publishers. There is no technical reason for this information asymmetry; it can be mitigated with better technology, in particular browser technology at the endpoint where all the data can be measured privately and confirmed anonymously.

The final requirement, which is only a soft requirement for Coasean analysis in the case of well-defined property rights, is that of low transaction costs. The Coasean transaction cost refers to the cost of negotiating a deal which can suit all parties to a dispute involving social costs. With the existing ecosystem, the transaction costs are impossibly high, with advertisers, publishers and users unable to come to terms.

In our example of present-day ad networks, we have a potential Coasean bargain between publishers and users, with a better outcome for advertisers as well. A Coasean solution to the attention economy inefficiencies for publishers and users is for advertisers to pay publishers by actual attention given to the publisher by the user.

Advertisers will pay the publisher for a share of the valuable attention the user pays to the publisher. Readers also will be directly compensated for their valued attention.

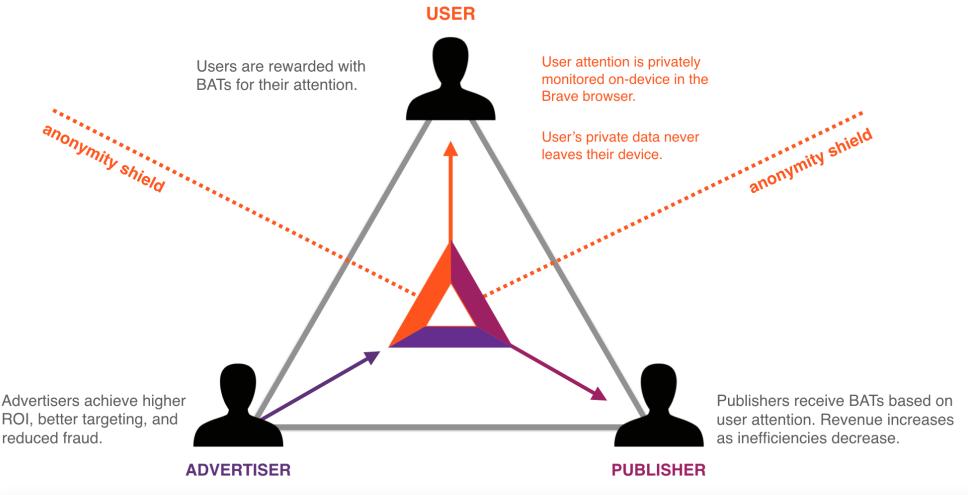


Figure 13: BAT Digital Ad Flow

The “pollution” of privacy invasiveness, slow browsing and data costs can be almost completely mitigated. Advertisers will know if their messages are delivered without having to resort to complex arguments about “viewability.” Publishers will not experience the negative externalities from the growing problem of ad blocker adoption.

Various proxies have been developed by advertisers and publishers to attempt to measure user attention using indirect techniques of “viewability,” but the advent of ad-blocking technologies and the increasing problem of fraud from non-human entities have cast doubt on such methods. A more direct technique would be to pay publishers via cryptographically secure methods, and serve the ad directly in the browser. Since the browser ultimately measures how the user interacts with the website better than any indirect meddling by intermediaries, involving the browser software itself in the process provides accurate measures of user attention bestowed on the publisher and advertiser.

The browser also provides a much richer data set for understanding what the individual user is interested in. The Brave browser will contain opt-in and transparent machine learning algorithms for assessing user interests. While an ad campaign targeted to a financial publisher may have value to the broad interests of the overall readership of the publisher, individual readers can be given ads tailored to their individual and even private preferences.

For example, sending an ad for discount bond brokers to people who are following the markets in municipal bond issues. The user who is reading about tech stocks and who has no interest in municipal bonds won’t receive the ad. The advertiser will effectively target the precise microsegment they are interested in reaching. The user receives more relevant ads while interest lasts, and private interests are not revealed to publishers or advertisers.

The idea that user attention should have monetary value is familiar to both publish-

ers and advertisers. The idea of publishers and particularly users being paid directly for attention bestowed on the publisher is novel. Since the valuable commodity is user attention, it makes economic sense that the user be compensated for their attention. One could justify this as a compensation for the externalities imposed on users by the advertising ecosystem. One could also justify this by the fact that one is more likely to perform an action if one is compensated for it. There is also confirmation that the actual user attention is bestowed on the publisher via the addition of cryptographic contracts built on blockchain to this advertising stack. The code is open source and can be reviewed by researchers and interested parties on the advertiser and publisher sides.

Since the transactions for the first deployment of BAT will happen through the Brave Ledger, which has privacy and deterministic user anonymity by design, full transparency can be achieved while user privacy is maintained. While this centralized solution should fulfill economic and technical goals, for further iterations, a decentralized solution could be developed to allow for trustless auditable transactions.

While paying a user to look at a publisher content may seem heretical to advertisers, the reality is the advertiser is paying someone. Removing the vast field of middlemen who add no value to the user/publisher relationship allows for a situation where the user may be compensated for valuable attention (made more valuable and relevant by measures of user interest at the browser) with no impact to advertiser costs and positive impact to publisher revenues. From a financial point of view, this could be seen as a variation on some other kind of short term promotion: advertisers regularly provide coupons and rebates on products. Promotions do not solve the problem of informing the user of the advertiser’s product in the first place. Promotions also don’t induce user loyalty or engagement. Most CMOs agree that short term sales can be improved with promotions, but sustainable competitive advantage can’t be achieved using promotions, hence the use of advertisements.

7.2 A Three-Way Coasean Bargain

The three-way Coase theorem is a source of much research interest among economists. The existence of “empty cores” in some situations have called into question the applicability of the Coase theorem to real world examples involving multiple distinct players[26]. While there are many more than three participants in the online ad market, we can idealize them as consisting of three participants: the advertiser, the publisher and the user. This analysis is useful for understanding the game theoretic considerations, for addressing any “empty core” arguments against the proposed Coasean bargain, as well as for illustrating the dire state of the publishing industry.

We propose the Basic Attention Token (BAT), a cryptographically-secure token, as the medium of exchange for facilitating this Coasean bargain while protecting the privacy of the user.

The advertiser wants to purchase user attention. This is broadly analogous to the “cost of production” in the exposition of the Coase theorem above, whose notation we follow.

The advertiser values the user attention with price C_a^a . The publisher wishes to

monetize the attention C_a^p paid to the website. The user who views the website values the content of the website with attention C_a^c .

Advertisers and publishers in the present ecosystem have transaction costs associated with monetization of attention. Publishers are paid by advertisers to provide user attention. The intermediaries of the present system create costs therefore $C_a^p < C_a^a$.

Note, when we talk about “transaction costs” apropos the Coase theorem, we refer to the transaction costs for negotiating a deal between the players of the Coasean game, therefore, rather awkwardly, the monetary costs of getting the ad to the publisher is not considered a “transaction cost” per se.

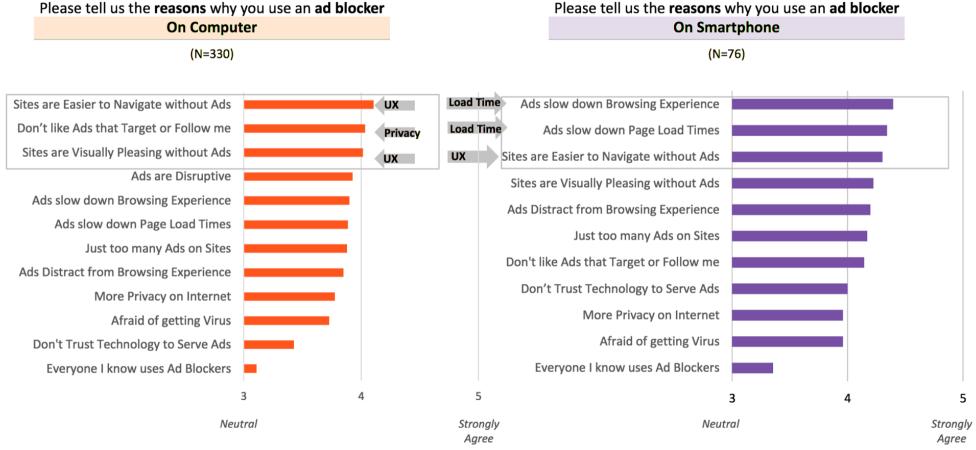
The present advertising ecosystem produces “social costs” or attention pollution as we have discussed above. These social costs are known to be large. For some large fraction of users (22% Lumascape state of the ad industry), the social costs are larger than the attention cost. We will label the pollution cost following the above example as P_a^c . In the present situation, the user will view the publisher and advertisers content so long as $C_a^c > P_a^c$. Every user is different, and of course, the publishers and advertisers vary as well, but the existence and growth of a large population of users for whom $C_a^c < P_a^c$ indicates that we are approaching the time where this inequality is always violated. The consequences of this are that $C_a^p = 0 \iff (C_a^c < P_a^c)$

Since C_a^c is proportional to Publisher profit (and advertiser profit in “attention”), any value which keeps $C_a^c > P_a^c$ is advantageous to the Publisher and Advertiser. Effectively the advertiser and the publisher combined are the factory in this argument, and the user owns the pollution rights. However, the user also values the product of the publisher. In the degenerate case where $C_a^c < P_a^c$ the user is also eventually harmed as the attention economy collapses, and the user takes up other hobbies.

The social cost should be decomposed into its constituent parts. We have identified the primary components of the social cost in our exposition of the advertising industry above. Security risk is one component, P^s . Hacker networks can place ads in irresponsible ad exchanges, which could have very large costs for individual users as well as the publisher who displays those ads.

Privacy loss is a very important social cost associated with the advertising landscape as it presently exists, P^p . Privacy invasions are presently required by advertisers to make sure the advertisement is actually viewed by a relevant user. In effect, the advertisers are paying for something which adds value to the attention.

Data costs are also a significant part of the social cost of the present day advertising ecosystem P^d . These costs are often borne by the user as a result of the activities of the middlemen who serve the advertiser and publisher. These costs seem most trivial, but for many users, they are among the top causes driving ad blocker adoption. For all viewers of online ad funded content, considerable time is taken in dealing with the cost of downloading and executing all the privacy-violating code. In addition to this cost, for those users who are using mobile devices, the monetary charges can be significant. It has been estimated that the top 50 news sites make 16 times less than the actual charges in data costs of delivering the advertising to the mobile user of these ads⁹! Since half or more of the data delivered by the publisher is advertising-related, half of a data plan can be hundreds of dollars a year in direct costs to the mobile user.



data source: IAB

Figure 14: Top Reasons to Block Ads: User Experience and Privacy

Finally, there is the cost to attention produced by the ad itself, P^a . In most cases, this is not a large cost, but as it is the thing actually valued most by advertisers, it should be accounted for separately. If ads can be made relevant, P^a may even be negative. Some users like looking at certain ads.

So, our total social cost for the present online ad ecosystem is

$$P_a^c = P^a + P^d + P^p + P^s$$

For a given value of P^a which is the thing actually valued by the advertiser, P_a^c will always be lower if we can eliminate the other factors. A token-based system with anonymizing features would remove P^p entirely. P^d will not be entirely mitigated by a token system, as some network traffic will take place to service the system and to present the ad itself. Since only a few bytes of data will need to be transferred to service the token, this cost will effectively only be in the downloading of the content of the advertisement; a considerable improvement. The use of cryptographic protocols and Zero Knowledge Proofs, as well as the use of known publishers and advertisers will also lower P^s considerably.

So, for a properly privacy protecting token system:

$$P_a^c(\text{BAT}) = P^a + P_{\text{BAT}}^d + P_{\text{BAT}}^s$$

To first order approximation,

$$P_a^c(\text{BAT}) \approx P^a + P_{\text{BAT}}^d$$

The remaining social cost can be reduced or eliminated by paying the user compensation which can be used for other things (for example, paying a publisher for premium

content or apps which relate to the content). In the simplified game-theoretic case presented here, the publisher eventually recovers this fraction of the ad spend anyway, since the publisher is the only place the attention token can be spent. In a more extensive case where the user can spend the tokens at other publishers, the revenues taken by the publisher are bounded by the ratio of the user’s take. The way tokens are apportioned in an advertising event in the proposed scheme, the publisher receives advertising spend that is much larger than the proportion of advertising spend they currently receive.

As the user also receives something which is of utility to him, we can safely declare that $P_a^c(\text{BAT})$ is zero or negative, which should encourage users to view more publisher content. Some may object that the token acquired by users for their attention activity can only be spent in the publisher’s “company store,” but as the token may be saved and used in different ways, it does have value to the user, just as airline points and video game tokens do.

The advertiser’s spending for a given amount of attention should be smaller in this ecosystem, since there are fewer social costs associated with delivering the required attention. In addition, the advertiser doesn’t ever have to pay for social cost to middlemen to achieve confidence their advertising content was shipped to a relevant user. Since this situation is better for publishers, and makes for happier and more “productive” users, advertisers should receive more benefit for their advertising spend.

To summarize, we have used the Coase theorem to demonstrate that the use of the BAT system offers lower costs to browser users, advertisers and publishers in the attention economy. Advertisers will receive a superior share of user attention, along with superior proof of user engagement. Publishers will receive a larger share of advertising revenues. Users will receive a superior experience with relevant ads and a share of advertising revenues.

7.3 An Analysis of the Stability of the BAT

A model for virtual currency exchange rates was postulated by Dutch economists von Oordt and Bolt in 2016[27]. The model postulates that the value of virtual currencies consists of three major factors; the utility of the virtual currency to make payments, the decision of forward-looking speculators to regulate the supply of virtual currency, and the elements that drive user adoption and merchant acceptance of a virtual currency.

The argument originates with Fisher’s 1911 observation that speculators may effectively limit the money supply by withdrawing money from circulation in anticipation of higher future utility. Since this dynamic particularly applies to limited issuance currencies such as bitcoin or BAT, it can be an important factor in the pricing for token sales and stability analysis of virtual currencies.

For a simple economic system with fixed quantity of currency tokens M^{BAT} , we can write down a transaction quantity relationship:

$$P_t^{\text{BAT}} T_t^{\text{BAT}} = M^{\text{BAT}} V_t^{\text{BAT}}$$

Where V_t^{BAT} is velocity of BAT, the average number of times each unit of BAT is used to purchase services within the defined period of time t . T_t^{BAT} is the quantity of

services purchased with BAT over the period of time t and P_t^{BAT} is the weighted price of the services.

Inserting the exchange rate in terms of \$

$$\frac{P_t^{\text{BAT}}}{P_t^{\$}} T_t^{\text{BAT}} = M^{\text{BAT}} V_t^{\text{BAT}}$$

Since we can assume the legacy fiat currency is the accounting unit for all parties involved, we define the exchange rate $S_t^{\frac{\$}{\text{BAT}}}$, and substitute in the above equation to give

$$S_t^{\frac{\$}{\text{BAT}}} = \frac{T_t^{\text{BAT}}}{M^{\text{BAT}} V_t^{\text{BAT}}}$$

If we consider the fraction of currency which is not used in transfer of services, we can postulate a velocity of the fraction of currency which is actually used for settlement V_t^{BAT} . Defining Z_t^{BAT} to be the number of BAT units not used in transactions.

Since the entire velocity of money in our economy V_t^{BAT} is an average between the currency units used and the units unused for transfer of services,

$$V_t^{\text{BAT}} = \frac{M^{\text{BAT}} - Z_t^{\text{BAT}}}{M^{\text{BAT}}} \widehat{V_t^{\text{BAT}}}$$

Combining these into the exchange rate

$$S_t^{\frac{\$}{\text{BAT}}} = \frac{\widehat{T_t^{\text{BAT}}}}{(M^{\text{BAT}} - Z_t^{\text{BAT}}) \widehat{V_t^{\text{BAT}}}} \quad (1)$$

The exchange rate for BAT tokens is therefore proportional to the volume of services purchased and inversely proportional to the currency not used in transactions for the time period t . This equation encapsulates the insight that a lack of money in circulation will raise the exchange rate.

We now turn our attention to the fraction of BAT which is not used for exchange. Some of the Z_t^{BAT} tokens may be the result of users forgetting about the small number of tokens they hold. Some may be due to exchange delays in settlement for legacy currencies. Overall though, the holders of inactive tokens have standard ways of evaluating future utility of the tokens in terms of modern risk management theory.

Since tokens do not bear interest, there is a discounted term associated with holding a position of size z_t^{BAT} in them.

$$-RS^{\frac{\$}{\text{BAT}}} z_t^{\text{BAT}}$$

where R is the interest rate discounting in the legacy currency.

If we consider the future expected value of the BAT holdings as the sum of the future expected value of the position in BAT

$$||S^{\frac{\$}{\text{BAT}}} t + 1|| z_t^{\text{BAT}}$$

with this discounted interest rate term (where R is the discounting operator), and the volatility of the future position in BAT scaled by a risk aversion term γ , we reach the efficient frontier from modern portfolio theory.

$$\|S_{t+1}^{\$}\|z_t^{\text{BAT}} - R(S_t^{\$})z_t^{\text{BAT}} + \gamma\sigma^2(\|S_{t+1}^{\$}\|)z_t^{\text{BAT}} = 0$$

Using this standard result, we can solve for the optimal number of tokens held by an individual during a given time period.

$$z_t^{\text{BAT}} = \frac{\|S_{t+1}^{\$}\| - R(S_t^{\$})}{\gamma\sigma^2(\|S_{t+1}^{\$}\|)}$$

If we consider all of the people holding BAT at a given time interval t we get the economically efficient number of BAT held for later use.

$$Z_t^{\text{BAT}} = N_t z_t^{\text{BAT}} = \frac{\|S_{t+1}^{\$}\|z_t^{\text{BAT}} - R(S_t^{\$})}{\frac{\gamma}{N_t}\sigma^2(\|S_{t+1}^{\$}\|)}$$

Since this value can't be negative, we assume that people who hold BAT have the position that

$$\|S_{t+1}^{\$}\| \geq R(S_t^{\$})$$

hence, using our above relationship, we get the relationship between the expected future value of the BAT, the interest rate and the velocity of transfers in the BAT economy:

$$R^{-1}(\|S_{t+1}^{\$}\|) \geq \frac{T_t^{\text{BAT}}}{M^{\text{BAT}}V_t^{\text{BAT}}}$$

So, people hold BAT if the discounted expected value exceeds the hypothetical value of the current exchange rate. So, the exchange rate as a function of future expected value of BAT is

$$S_t^{\$} = R^{-1}(\|S_{t+1}^{\$}\|) - \frac{\gamma}{N_t} Z_t^{\text{BAT}} \sigma^2(\|S_{t+1}^{\$}\|) \quad (2)$$

Thus, the BAT holdings are the discounted expected future exchange rate minus the risk premium for the uncertainty in future value of the BAT.

If the model holds, 1 and 2 can be used to define supply and demand for BAT. Since M^{BAT} is not time dependent in the case of BAT, the time varying exchange rate can be readily understood in terms of BAT transactions and opinions on future utility of BAT transactions. As BAT transactions increase, the exchange rate becomes dominated by the transactions rather than future expectations of utility. This dynamic has been observed in maturing virtual currencies as well as various other in-house token systems.

While models are imprecise, this model argues for long term price stability in a token mediated economy.

References

- [1] MIT Technology Review and Vigilant. "Navigating Planet Ad Tech: A Guide for Marketers". In: *MIT Technology Review* (Oct. 2013). URL: <https://www.technologyreview.com/s/519991/navigating-planet-ad-tech/>.
- [2] T. H.; Beck J. C. Davenport. *The Attention Economy: Understanding the New Currency of Business*. Harvard Business School Press, 2001. ISBN: 978-1578514410.
- [3] Wikipedia. *AIDA (marketing)*. [Online; accessed 22-January-2017]. 2017. URL: [https://en.wikipedia.org/wiki/AIDA_\(marketing\)](https://en.wikipedia.org/wiki/AIDA_(marketing)).
- [4] Jack Neff. "P&G Tells Digital to Clean Up, Lays Down New Rules for Agencies and Ad Tech to Get Paid". In: *Advertising Age* (Jan. 2017). URL: <http://adage.com/article/media/p-g-s-pritchard-calls-digital-grow-up-new-rules/307742/>.
- [5] Paul Sholtz. "Transaction Costs and the Social Costs of Online Privacy". In: *First Monday* 6.5 (May 2001). URL: http://firstmonday.org/issues/issue6_5/sholtz/index.html.
- [6] Lee Rainie. "The state of privacy in post-Snowden America". In: *Pew Research Center FactTank* (Sept. 2016). URL: <http://www.pewresearch.org/fact-tank/2016/09/21/the-state-of-privacy-in-america/>.
- [7] Margaret Boland. *Cyber criminals are stealing billions from the ad industry each year*. [Online; accessed 22-January-2017]. 2016. URL: <http://www.businessinsider.com/the-ad-fraud-report-bot-traffic-2016-3>.
- [8] Hillary Tuttle. "The Rise of Malvertising". In: *Risk Management Monitor* (Aug. 2015). URL: <http://www.riskmanagementmonitor.com/the-rise-of-malvertising/>.
- [9] Rob Leathern. "Carriers are Making More From Mobile Ads than Publishers Are". In: *Medium* (Oct. 2015). URL: <https://medium.com/@robleathern/carriers-are-making-more-from-mobile-ads-than-publishers-are-d5d3c0827b39#.aiw3hs4ls>.
- [10] eMarketer. *US Ad Blocking to Jump by Double Digits This Year*. [Online; accessed 22-January-2017]. June 2016. URL: <https://www.emarketer.com/Article/US-Ad-Blocking-Jump-by-Double-Digits-This-Year/1014111>.
- [11] Interactive Advertising Bureau. *Ad Blocking: Who Blocks Ads, Why and How to Win Them Back*. Tech. rep. Interactive Advertising Bureau, 2016. URL: <http://www.iab.com/wp-content/uploads/2016/07/IAB-Ad-Blocking-2016-Who-Blocks-Ads-Why-and-How-to-Win-Them-Back.pdf>.
- [12] Mathew Ingram. "How Google and Facebook Have Taken Over the Digital Ad Industry". In: *Fortune* (Jan. 2017). URL: <http://fortune.com/2017/01/04/google-facebook-ad-industry/>.
- [13] Mark Jurkowitz Amy Mitchell and Kenneth Olmstead. *Social, Search and Direct: Pathways to Digital News*. Tech. rep. Pew Research Center, Mar. 2014. URL: <http://www.journalism.org/2014/03/13/social-search-direct/>.

- [14] Digital Content Next Research Team. *DCNs Distributed Content Revenue Benchmark Report*. Tech. rep. Digital Content Next, Jan. 2017. URL: <https://digitalcontentnext.org/blog/2017/01/25/dcns-distributed-content-revenue-benchmark-report/>.
- [15] YouExec. *Google & Facebook ad traffic is 90% useless*. [Online; accessed 22-January-2017]. Jan. 2017. URL: <https://youexec.com/dev/2017/1/14/google-facebook-ads-traffic-is-useless>.
- [16] Chris Pemberton. *Gartner CMO Spend Survey 2016-2017 Shows Marketing Budgets Continue to Climb*. Tech. rep. Gartner Research, Dec. 2016. URL: <https://www.gartner.com/smarterwithgartner/gartner-cmo-spend-survey-2016-2017-shows-marketing-budgets-continue-to-climb/>.
- [17] Jack Simpson. *40% of publishers describe their digital ad revenue as shrinking or static*. Tech. rep. Econsultancy, Oct. 2015. URL: <https://econsultancy.com/blog/67028-40-of-publishers-describe-their-digital-ad-revenue-as-shrinking-or-static/>.
- [18] Dimitri DeFigueiredo. *Github discussion of concave score*. May 2016. URL: <https://github.com/brave/ledger/issues/2#issuecomment-221752002>.
- [19] S. Myers R. Pass S. Hohenberger and A. Shelat. “An Overview of ANONIZE: A Large-Scale Anonymous Survey System”. In: *IEEE Security and Privacy* 13.2 (2015), pp. 22–29.
- [20] Abhi Shelat Rafael Pass. “Micropayments for Decentralized Currencies”. In: *CCS '15: Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security* (2015), pp. 207–218.
- [21] Matthew D. Green Jingcheng Liu Ian Miers Peihan Miao Pratyush Mishra Alessandro Chiesa. “Decentralized Anonymous Micropayments”. In: *EUROCRYPT 2017 (36th International Conference on the Theory and Applications of Cryptographic Techniques)* (2017).
- [22] Ian Miers Matthew Green. “Bolt: Anonymous Payment Channels for Decentralized Currencies”. In: *IACR Cryptology ePrint Archive* 2016 (2016).
- [23] Jens Groth. “Short pairing-based non-interactive zero-knowledge arguments”. In: *Proceedings of the 16th International Conference on the Theory and Application of Cryptology and Information Security, ASIACRYPT '10* (2010), pp. 321–340.
- [24] Iddo Ben-Tov Alessandro Chiesa Ariel Gabizon Daniel Genkin Matan Hamilis Evgenya Pergament Michael Riabzev Mark Silberstein Eran Tromer Eli Ben-Sasson and Madars Virza. “Computational integrity with a public random string from quasi-linear PCPs”. In: *EUROCRYPT 2017 (36th International Conference on the Theory and Applications of Cryptographic Techniques)* (2017).

- [25] Reed Hundt. *Statement of Reed Hundt, Chairman of the Federal Communications Commission on Spectrum Policy Management before the Subcommittee on Telecommunications, Trade and user Protection, Committee on Commerce, U.S. House of Representatives*. Feb. 1997. URL: <https://transition.fcc.gov/Speeches/Hundt/spreh743.html>.
- [26] J. Callen V. Aivazian. “The Coase Theorem and the Empty Core”. In: *Journal of Law and Economics* 24 (1 1981), pp. 175–181.
- [27] Wilko Bolt and Maarten van Oordt. *On the Value of Virtual Currencies*. Tech. rep. Working Paper No. 2016-42. Bank of Canada, Apr. 2016.

D:20180313001459Z



Binance Exchange

www.binance.com

Whitepaper
V1.2

Intro	3
Problems	3
Binance Exchange	4
Matching Engine	4
Feature Rollout	4
Coins	5
Device Coverage	5
Multilingual Support	5
UI Preview	6
Revenue Model	7
Binance Coin (BNB)	7
Allocation	8
ICO	8
ICO Schedule	8
BNB Value & The Burn	9
BNB Vesting Plan for the Team	9
Funds Usage	10
Team	10
Changpeng Zhao - CEO	10
Roger Wang - CTO	11
James Hofbauer - Chief Architect	12
Paul Jankunas - VP of Engineering	12
Allan Yan - Product Director	13
Sunny Li - Operations Director	13
Investors & Advisors	14
Risks	17
Security is Paramount	17
Market Competition	17

Intro

In our view, there are fundamentally two different types of exchanges: the ones that deal with fiat currency; and the ones that deal purely in crypto. It is the latter one that we will focus on. Even though they are small now, we strongly believe that pure crypto exchanges will be bigger, many times bigger, than fiat based exchanges in the near future. They will play an ever more important role in world finance and we call this new paradigm **Binance**; Binary Finance.

With your help, **Binance** will build a world-class crypto exchange, powering the future of crypto finance.

Problems

Some of the current crypto exchanges suffer from a number of problems:

- **Poor technical architecture**

Many exchanges are “put together quickly”, by good tech people, but who have little or no experience in finance or in operating an exchange. They often choose the simplest approach to get the system up and running. While this may work well in the beginning, as traffic grows, the system will not be able to handle the increased load. Exchange systems need to be engineered from the ground up with security, efficiency, speed, and scalability in mind. This often slows down the initial development, but is critical for long-term success.

Our team has decades of combined experience building and maintaining world class financial systems that shape the economy. We understand how these systems are built from the ground up.

- **Insecure platform**

There are hundreds of exchanges that went down due to being hacked¹.

Binance is built to high standards, audited, and penetration tested. We have experience building financial systems to the highest security standards and strive to ensure security first.

- **Poor market liquidity**

Professional traders and normal users are significantly affected by this. Having a shallow orderbook means high slippage when trading, which is very expensive for traders. Getting miners, institutional investors and large traders into a new exchange is a chicken and egg problem, and requires a team with deep industry resources.

¹ <https://bitcointalk.org/index.php?topic=576337>

Binance's team have been in both the finance and crypto industry for many years. The team has worked on and operated a number of exchanges, and have accumulated a large network of partners in this space. These partners will be key in bootstrapping the exchange.

- **Poor customer service**

Traders are a different breed when it comes to users. Understanding the trader mentality is vital for running a successful exchange. Money is literally on-the-line. Many exchanges service traders as if they were running a social media site. A 3-second delay in seeing your friends' status update would hardly be noticed, but on an exchange, the same would be unacceptable, resulting in a torrent of user complaints.

In addition to the technology stack, Binance is built with service in mind. Binance shares support responsibilities across the entire staff and company. When a trader has a problem, they get an answer directly from someone who knows the system and not someone reading from a script.

- **Poor internationalization and language support**

Blockchains have no borders. Most exchanges focus only on one language or one country.

Our international multi-lingual team has extensive working experience in North America, Europe and Asia, and we are able to smoothly support the global market.

Binance Exchange

Matching Engine

Our matching engine is capable of sustaining 1,400,000 orders / second, making Binance one of the fastest exchanges in the market today. You can be certain, on our exchange, that your orders will never be stuck due to the matching engine being overwhelmed.

Feature Rollout

We will roll out the platform in roughly the following order:

- Spot trading
- Margin trading
- Futures
- Anonymous instant exchange
- Decentralized (on-chain) exchange
- and more...

Coins

Binance will support trading pairs in the following coins:

- BTC
- ETH
- LTC
- NEO (ANS)
- BNB (Binance Coin)

More coins will be added over time. We generally will only add coins that have strong credibility, user base, and liquidity. If you have a coin that you wish to be listed on Binance later, participating in our ICO will help.

We have no plans to support any fiat currencies such as USD, RMB, JPY, or KRX.

Device Coverage

We will provide cross-platform trading clients for:

- Web-based trading client
- Android native client
- iOS native client (pending App Store review)
- Mobile HTML5 client (including WeChat H5 client)
- PC (Windows) native client
- REST API

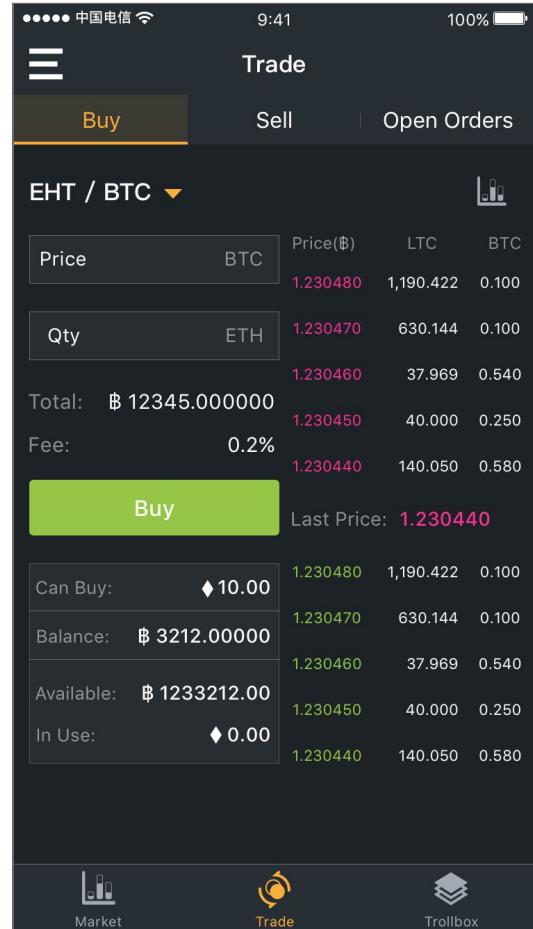
Multilingual Support

We will support English, Chinese, Japanese and Korean on all of our user interfaces. (The very initial release will be in English and Chinese only.) More languages will be added over time.

UI Preview

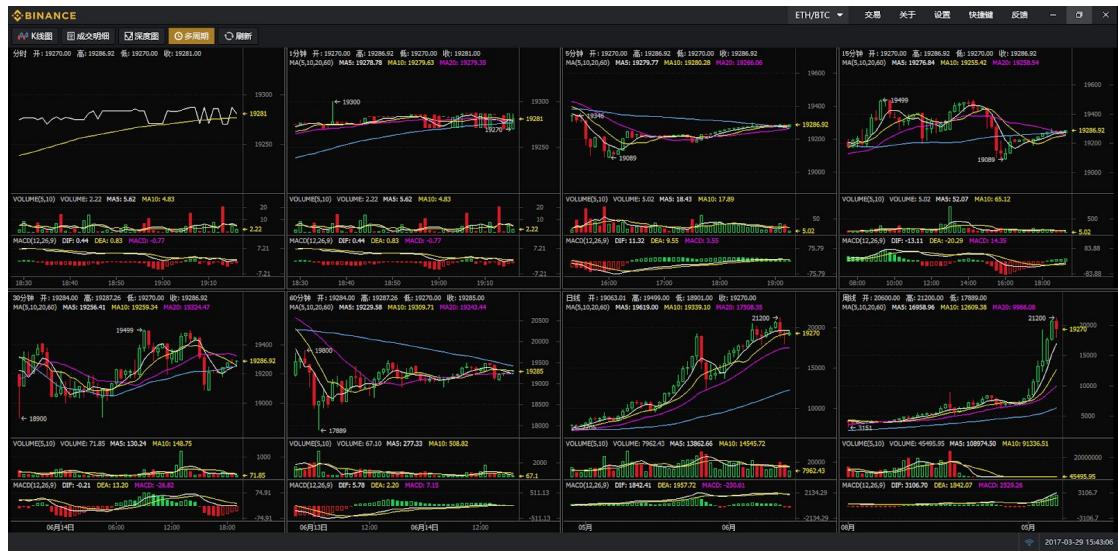


Professional Web Trading Interface



Mobile HTML5 Market Data

iOS Trading Page



Windows PC Native Client - Multi-Interval View

Revenue Model

Binance's revenue will come from the following sources:

Source	Description
Exchange Fee	Binance initially will charge a 0.2% fixed fee per trade. Other variations will be subsequently introduced, including maker-taker, volumed based tiering and 0 fee promotions. We have no plan to charge above 0.2%.
Withdrawal Fee	Binance may charge a small fee for withdrawals.
Listing Fee	Binance will select innovative coins and other assets to be listed on the exchange, there may be a fee associated with those listings.
Margin Fee	If you trade on margin, there may be a fee or interest on the borrowed amount.
Other Fees	There may be other fees the platform may collect for various services such as automated algorithmic order etc.

Binance Coin (BNB)

We will issue our token coin, called the Binance Coin. A strict limit of 200MM BNB will be created, never to be increased. BNB will run natively on the Ethereum blockchain with ERC 20.

Allocation

%	Amount (BNB)	Participant
50%	100,000,000	ICO
40%	80,000,000	Founding Team
10%	20,000,000	Angel investors

ICO

The ICO will be done in BTC and ETH, on multiple platforms around the world.

ICO Schedule

All times below are China Standard Time (CST), UTC+8 hours.

Date	Task
2017/06/14	Confirmed start of the Binance project
2017/06/16	Initial draft white paper completed, circulated to potential angel investors
2017/06/22	Announce Binance ICO plan, and release whitepaper to general public
2017/07/01	ICO starts (platforms will be announced soon)
2017/07/15	Binance.com release v0.1 go live, active trading begins
2017/07/21	ICO finishes, or whenever the coins are sold out

ICO will start from 3PM July 1st, investors can purchase BNB tokens in 3 phases on a first-come, first-served basis until 100,000,000 tokens are sold. As each new phase starts, the price will increase.

Investors will receive BNB tokens within 5 working days after the ICO finishes. The detailed schedule is as below:

ICO Phase	1st week	2nd week	3rd week
-----------	----------	----------	----------

CST/GMT+8	15:00 July 1th - 15:00 July 7th	15:00 July 7th - 15:00 July 14th	15:00 July 14th - 15:00 July 21th
1ETH	2700 BNB	2500 BNB	2300 BNB
1BTC	Based on market price		

BNB Value & The Burn

You can use BNB to pay for any fees on our platform, including but not limited to:

- Exchange fees
- Withdraw fees
- Listing fees
- Any other fee

When you use BNB to pay for fees, you will receive a significant discount:

	1st year	2nd year	3rd year	4th year	5th year
Discount Rate	50%	25%	12.5%	6.75%	no discount

The Burn

Every quarter, we will destroy BNB based on the trading volume on our crypto-to-crypto platform until we destroy 50% of all the BNB. All transactions will be on the blockchain. We eventually will destroy 100MM BNB, leaving 100MM BNB remaining.

Decentralized Exchange

In the future, Binance will build a decentralized exchange, where BNB will be used as one of the key base assets as well as gas to be spent.

BNB Vesting Plan for the Team

Initial release:	20% (16MM)
After 1 year:	20% (16MM)
After 2 year:	20% (16MM)
After 3 year:	20% (16MM)
After 4 year:	20% (16MM)

Funds Usage

- 35% of the funds will be used to build the Binance platform and perform upgrades to the system, which includes team recruiting, training, and the development budget.
- 50% will be used for Binance branding and marketing, including continuous promotion and education of Binance and blockchain innovations in industry mediums. A sufficient budget for various advertisement activities, to help Binance become popular among investors, and to attract active users to the platform.
- 15% will be kept in reserve to cope with any emergency or unexpected situation that might come up.

Team

We have a solid team led by Changpeng Zhao, with both traditional wall street finance and cryptocurrency experience. We have a track record of successful startups under our belt.

Changpeng Zhao - CEO

(aka CZ in the crypto community) [LinkedIn Profile](#)



CZ is the founder and CEO of BijieTech, a company that provides cloud-based exchange systems to exchange operators. Since founding in Sept 2015, BijieTech now powers 30+ exchanges in Asia. In the first 12 months since founding, BijieTech closed 36.1 million RMB (\$5.3MM USD) in revenue, and will double that in its second year. BijieTech has never accepted any outside investments, being cash flow positive from day one.

As soon as the Binance ICO finishes, CZ will remain a shareholder of BijieTech, but will relinquish all of his management duties to a new CEO. CZ will focus exclusively on Binance. This applies to all BijieTech members listed in this whitepaper.

Prior to BijieTech, CZ was the co-founder and CTO of OKCoin. During his stay there, OKCoin launched their international site, and their futures trading platform. Co-ordinating with Stefan Thomas, CZ also lead the first proof-of-reserves in any China crypto exchange. Most other major exchanges in China followed soon after. In addition to managing the tech team there, he also lead the international marketing team. He is still mentor to and good friends with Zane Tackett.

Before OKCoin, CZ was the Head of Technology and the 3rd person to join the Blockchain.info team. He worked closely with Ben Reeves, Roger Ver, Anthony Antonopoulos and Nicolas Cary to grow the Blockchain.info service.

Before Blockchain.info, CZ co-founded Fusion Systems Ltd in 2005, a company that specializes in ultra-low-latency trading systems for brokers. Fusion Systems was started in Shanghai, and currently has offices in Tokyo, Hong Kong, and Los Angeles. Among other tasks, CZ was responsible closing and deploying trading systems at Credit Suisse, Goldman Sachs, Deutsche Bank, and more. CZ left Fusion System to work full time in the blockchain industry in 2013.

Before Fusion Systems, CZ was the Head of Development at Bloomberg Tradebook Futures for 4 years, in New York. There CZ managed a team that was responsible for the entire futures trading platform in Bloomberg, with annual revenues exceeding \$300 million USD.

Prior to Bloomberg, CZ's college internship and first job out of college was in Tokyo, working for a tech outsource company that was involved in developing trading systems for the Tokyo Stock Exchange. This is where his exchange experience began.

CZ was born in China and went to high school and college in Canada. CZ is fully bilingual in English and Chinese, and can speak basic Japanese.

Roger Wang - CTO

[LinkedIn Profile](#)



Roger is a co-founder and the CTO. He has been working in the financial industry for 10+ years, responsible for building up technical teams, designing the high level architecture of the exchange and clearing systems, and running ops teams to ensure the security and stability of exchange systems.

Prior to BijieTech, Roger worked at Nomura Securities, the largest investment bank in Japan. He was responsible for a global credit booking, analytics, and marking system, which supported thousands of global traders and analysts. He has also successfully implemented a smart bond matching engine, which consumes firm wide trading/order/position data, client enquiry info, 3rd party data as well as public bond data, to discover business opportunities for the firm using sophisticated custom built algorithms. It now generates over 100MM USD revenue annually for Nomura.

Before Nomura, he was a tech leader in Morgan Stanley, where he designed and built a financial TB level data warehouse, which supported a large number of users to

do real time data analysis and financial modeling. He was also a core developer for low latency algorithmic trading systems, which served the firm's largest clients like Blackstone and Wellington Fund, that delivered significant commission income for Morgan Stanley.

Roger is fully trilingual in English, Chinese and Japanese.

James Hofbauer - Chief Architect

[LinkedIn Profile](#)



James is a co-founder and the Chief Architect of BijieTech. He architects and implements the core matching engine and its middleware. He also oversees client exchanges' public endpoints to ensure security and high performance.

Before BijieTech, James worked at Palantir, a Silicon Valley company that focuses on big data analysis. Palantir's large-scale high-performance systems are used for cyber-security, anti-money laundering, fraud detection, counter-terrorism, and many other data relationship analysis purposes by both private and government entities.

Before Palantir, James worked at Fusion Systems. A notable project James worked on was a global investment bank's systems architecture redesign, focusing on reducing the number of systems, encrypting and securely handling sensitive data, and introduced a new user security system which provides authentication, authorization, and action audit logging.

James was raised and educated in America, earning a Bachelor's of Science, *Cum Laude*, in Computer Science. He is bilingual, native in English and fluent in Japanese (JLPT N2 certified), and has lived in Japan for over 10 years.

James has known CZ for 7 years and they have worked in two startups together.

Paul Jankunas - VP of Engineering

[LinkedIn Profile](#)



Paul is the VP of Engineering at BijieTech, responsible for the C++ implementation of the core machine engine. He has over 15 years of experience in developing exchange systems and financial trading applications. He is constantly looking for new ways to improve the performance and scalability of the system.

Prior to BijieTech, Paul worked at SBI BITS, part of SBI Group, in Tokyo. SBI Group is a listed financial services company with interests in a wide assortment of businesses. Paul was responsible for both client and server side development for trading applications.

Before that, Paul worked at Fusions Systems in Tokyo as the Head of Development on Raptor, a market gateway with latencies under 2 microseconds, and before that for Bloomberg in New York.

Paul has known CZ for 9 years and they have worked together in 3 companies.

Allan Yan - Product Director

[LinkedIn Profile](#)



Allan is a co-founder and the Product Director of BijieTech. Allan has over 10 years of experience in product design, user experience and trading. He drives the innovations in the exchange systems built by BijieTech, and pushes the product far ahead of the competition in this ultra competitive space.

Before BijieTech, Allan worked in Orient International Holding, which is one of the biggest import & export firms in Shanghai. He was responsible for the implementation of several informationization products, including ERP and e-Fax. Meanwhile, he led a variety of game and VOD content platforms.

Sunny Li - Operations Director

[LinkedIn Profile](#)



Sunny is a co-founder and the Operations Director of BijieTech. He has many years of management and technology consulting experience, has led 20+ exchange systems projects, and provided comprehensive consulting for strategy, operations, risk control and system development.

Prior to BijieTech, Sunny worked at Accenture as the senior consultant. He provided many Top 500 companies for strategic and IT consulting, and led a number of IOT, big data, ERP information integration systems projects.

Investors & Advisors

In no particular order.



Matthew Roszak

Bloq co-founder.
Tally Capital Founding
Partner.



Roger Ver

Angel investors in many
blockchain businesses.
CEO of Bitcoin.com.



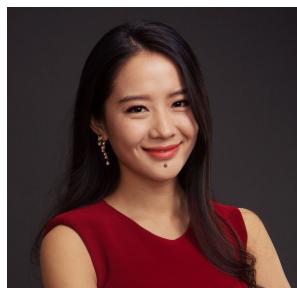
Ron Cao

MD of Sky9 Capital
Institutional investor in
BTCChina.



Chandler Guo

Angel Investor in
blockchain businesses in
China.



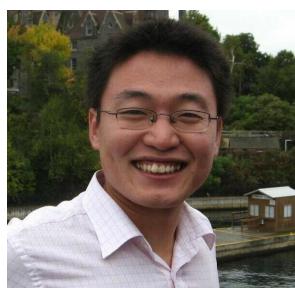
He Yi

CMO of Yixia
Technology.
Previously Co-founder at
OKCoin.



Yang Linke

Co-founder of BTCChina.
ICOCoin Founder.



Zhao Dong

One of the largest crypto
OTC brokers in China.



Da Hongfei

AntShares Founder.
Onchain CEO.



Jun Du

Co-founder of Huobi.
Angel investor.



Vincent Zhou



Lu Bin



Liu Sutong

Founder of FinTech Blockchain Group. Active angel blockchain investor.

CEO of Andui.com, a blockchain financial service company in China

Finance Channel TV Anchor.
CEO of Heng Pool.



Eric Zhang

AntShares Core Member.
Lead match-engine developer at Huobi.com



Leah Zhang

CMO of F2Pool.
Previously Investment Manager of AngelCrunch.



Wang Qijun

Co-founder of Andui.
Formerly Marketing at Blockchain.info



Roy Zou

CEO of Bitkio, Secretarl at Ethereum Classic Consortium (ECC)



Jackie Wang

Founding team member of Bitbank.com and BW.com and CHBTC.



Li Da

Co-founder of JiulianTech.



Xiaoning Nan

Founder of BitOcean.



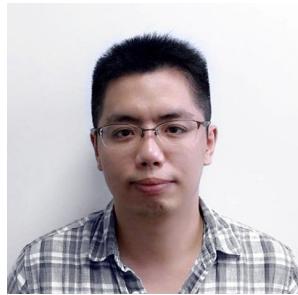
Jeff Cui

Founder and CEO of TKing.cn. Tech lead at Morgan Stanley.



Guicheng Xiong

Co-founder of 91 Wireless, acquired by Baidu at \$1.9 billion USD.



Xin Chen

Previous Product Director of OKCoin. Analyst at Guotai Junan Securities.



William Liu

Senior Partner at AllBright Law Offices, the biggest Law Firm in Shanghai.

Some investors choose to remain private.

Risks

There are many risks involved in running an exchange. We understand this and have the skills, experience, and leadership to overcome them.

Security is Paramount

Many crypto exchanges have failed due to poor security procedures. Most security breaches could have been prevented by taking simple precautions to protect critical resources. Our team has developed Binance with security as the foremost concern in their minds. We strive to ensure that we have followed all the industry best practices when it comes to securing infrastructure and data including ISO/IEC 27001:2013² and the CryptoCurrency Security Standard (CCSS)³.

Market Competition

We know this will be an ultra competitive space. There are probably hundreds, if not thousands of teams wanting, planning or doing exchanges. Competition will be fierce. But in this age, this is a common risk in any decent concept/startup or mature company. The question is: given our team, track record, experience, industry resources, and product, do you believe we stand a better chance than the rest of the pack? If yes, then please join our ICO.

² https://en.wikipedia.org/wiki/ISO/IEC_27001:2013

³ <https://cryptoconsortium.org/standards/CCSS>

Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

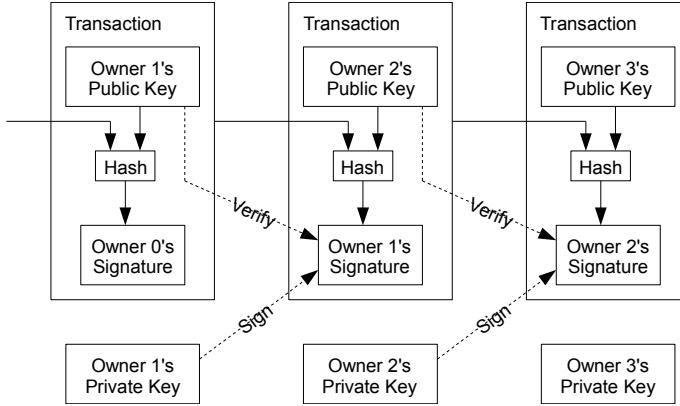
1. Introduction

Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model. Completely non-reversible transactions are not really possible, since financial institutions cannot avoid mediating disputes. The cost of mediation increases transaction costs, limiting the minimum practical transaction size and cutting off the possibility for small casual transactions, and there is a broader cost in the loss of ability to make non-reversible payments for non-reversible services. With the possibility of reversal, the need for trust spreads. Merchants must be wary of their customers, hassling them for more information than they would otherwise need. A certain percentage of fraud is accepted as unavoidable. These costs and payment uncertainties can be avoided in person by using physical currency, but no mechanism exists to make payments over a communications channel without a trusted party.

What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party. Transactions that are computationally impractical to reverse would protect sellers from fraud, and routine escrow mechanisms could easily be implemented to protect buyers. In this paper, we propose a solution to the double-spending problem using a peer-to-peer distributed timestamp server to generate computational proof of the chronological order of transactions. The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes.

2. Transactions

We define an electronic coin as a chain of digital signatures. Each owner transfers the coin to the next by digitally signing a hash of the previous transaction and the public key of the next owner and adding these to the end of the coin. A payee can verify the signatures to verify the chain of ownership.

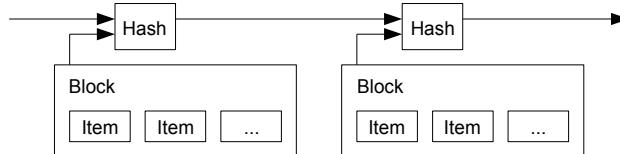


The problem of course is the payee can't verify that one of the owners did not double-spend the coin. A common solution is to introduce a trusted central authority, or mint, that checks every transaction for double spending. After each transaction, the coin must be returned to the mint to issue a new coin, and only coins issued directly from the mint are trusted not to be double-spent. The problem with this solution is that the fate of the entire money system depends on the company running the mint, with every transaction having to go through them, just like a bank.

We need a way for the payee to know that the previous owners did not sign any earlier transactions. For our purposes, the earliest transaction is the one that counts, so we don't care about later attempts to double-spend. The only way to confirm the absence of a transaction is to be aware of all transactions. In the mint based model, the mint was aware of all transactions and decided which arrived first. To accomplish this without a trusted party, transactions must be publicly announced [1], and we need a system for participants to agree on a single history of the order in which they were received. The payee needs proof that at the time of each transaction, the majority of nodes agreed it was the first received.

3. Timestamp Server

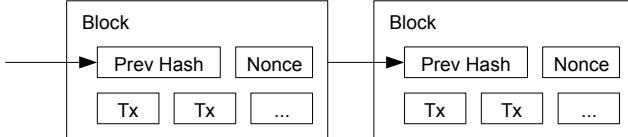
The solution we propose begins with a timestamp server. A timestamp server works by taking a hash of a block of items to be timestamped and widely publishing the hash, such as in a newspaper or Usenet post [2-5]. The timestamp proves that the data must have existed at the time, obviously, in order to get into the hash. Each timestamp includes the previous timestamp in its hash, forming a chain, with each additional timestamp reinforcing the ones before it.



4. Proof-of-Work

To implement a distributed timestamp server on a peer-to-peer basis, we will need to use a proof-of-work system similar to Adam Back's Hashcash [6], rather than newspaper or Usenet posts. The proof-of-work involves scanning for a value that when hashed, such as with SHA-256, the hash begins with a number of zero bits. The average work required is exponential in the number of zero bits required and can be verified by executing a single hash.

For our timestamp network, we implement the proof-of-work by incrementing a nonce in the block until a value is found that gives the block's hash the required zero bits. Once the CPU effort has been expended to make it satisfy the proof-of-work, the block cannot be changed without redoing the work. As later blocks are chained after it, the work to change the block would include redoing all the blocks after it.



The proof-of-work also solves the problem of determining representation in majority decision making. If the majority were based on one-IP-address-one-vote, it could be subverted by anyone able to allocate many IPs. Proof-of-work is essentially one-CPU-one-vote. The majority decision is represented by the longest chain, which has the greatest proof-of-work effort invested in it. If a majority of CPU power is controlled by honest nodes, the honest chain will grow the fastest and outpace any competing chains. To modify a past block, an attacker would have to redo the proof-of-work of the block and all blocks after it and then catch up with and surpass the work of the honest nodes. We will show later that the probability of a slower attacker catching up diminishes exponentially as subsequent blocks are added.

To compensate for increasing hardware speed and varying interest in running nodes over time, the proof-of-work difficulty is determined by a moving average targeting an average number of blocks per hour. If they're generated too fast, the difficulty increases.

5. Network

The steps to run the network are as follows:

- 1) New transactions are broadcast to all nodes.
- 2) Each node collects new transactions into a block.
- 3) Each node works on finding a difficult proof-of-work for its block.
- 4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
- 5) Nodes accept the block only if all transactions in it are valid and not already spent.
- 6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Nodes always consider the longest chain to be the correct one and will keep working on extending it. If two nodes broadcast different versions of the next block simultaneously, some nodes may receive one or the other first. In that case, they work on the first one they received, but save the other branch in case it becomes longer. The tie will be broken when the next proof-of-work is found and one branch becomes longer; the nodes that were working on the other branch will then switch to the longer one.

New transaction broadcasts do not necessarily need to reach all nodes. As long as they reach many nodes, they will get into a block before long. Block broadcasts are also tolerant of dropped messages. If a node does not receive a block, it will request it when it receives the next block and realizes it missed one.

6. Incentive

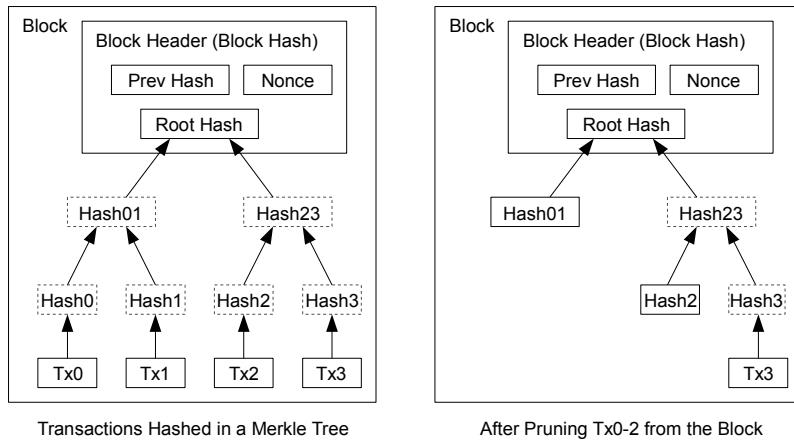
By convention, the first transaction in a block is a special transaction that starts a new coin owned by the creator of the block. This adds an incentive for nodes to support the network, and provides a way to initially distribute coins into circulation, since there is no central authority to issue them. The steady addition of a constant of amount of new coins is analogous to gold miners expending resources to add gold to circulation. In our case, it is CPU time and electricity that is expended.

The incentive can also be funded with transaction fees. If the output value of a transaction is less than its input value, the difference is a transaction fee that is added to the incentive value of the block containing the transaction. Once a predetermined number of coins have entered circulation, the incentive can transition entirely to transaction fees and be completely inflation free.

The incentive may help encourage nodes to stay honest. If a greedy attacker is able to assemble more CPU power than all the honest nodes, he would have to choose between using it to defraud people by stealing back his payments, or using it to generate new coins. He ought to find it more profitable to play by the rules, such rules that favour him with more new coins than everyone else combined, than to undermine the system and the validity of his own wealth.

7. Reclaiming Disk Space

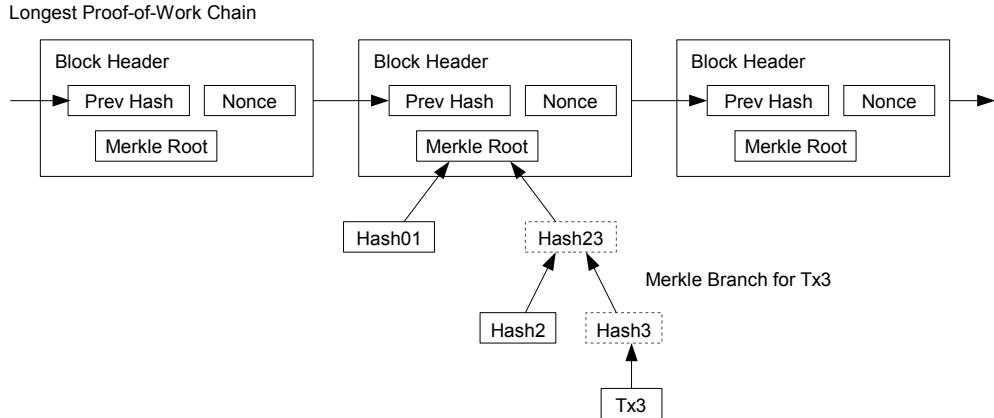
Once the latest transaction in a coin is buried under enough blocks, the spent transactions before it can be discarded to save disk space. To facilitate this without breaking the block's hash, transactions are hashed in a Merkle Tree [7][2][5], with only the root included in the block's hash. Old blocks can then be compacted by stubbing off branches of the tree. The interior hashes do not need to be stored.



A block header with no transactions would be about 80 bytes. If we suppose blocks are generated every 10 minutes, $80 \text{ bytes} * 6 * 24 * 365 = 4.2\text{MB}$ per year. With computer systems typically selling with 2GB of RAM as of 2008, and Moore's Law predicting current growth of 1.2GB per year, storage should not be a problem even if the block headers must be kept in memory.

8. Simplified Payment Verification

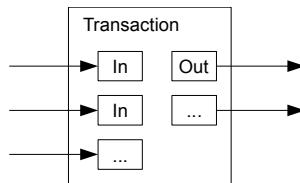
It is possible to verify payments without running a full network node. A user only needs to keep a copy of the block headers of the longest proof-of-work chain, which he can get by querying network nodes until he's convinced he has the longest chain, and obtain the Merkle branch linking the transaction to the block it's timestamped in. He can't check the transaction for himself, but by linking it to a place in the chain, he can see that a network node has accepted it, and blocks added after it further confirm the network has accepted it.



As such, the verification is reliable as long as honest nodes control the network, but is more vulnerable if the network is overpowered by an attacker. While network nodes can verify transactions for themselves, the simplified method can be fooled by an attacker's fabricated transactions for as long as the attacker can continue to overpower the network. One strategy to protect against this would be to accept alerts from network nodes when they detect an invalid block, prompting the user's software to download the full block and alerted transactions to confirm the inconsistency. Businesses that receive frequent payments will probably still want to run their own nodes for more independent security and quicker verification.

9. Combining and Splitting Value

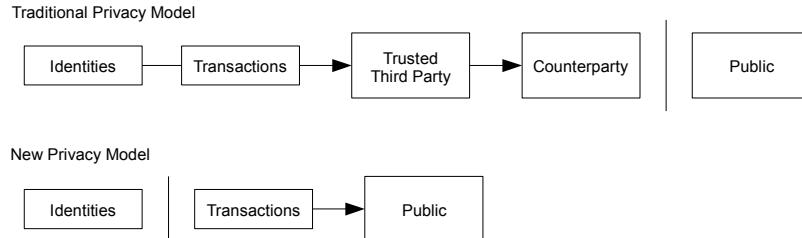
Although it would be possible to handle coins individually, it would be unwieldy to make a separate transaction for every cent in a transfer. To allow value to be split and combined, transactions contain multiple inputs and outputs. Normally there will be either a single input from a larger previous transaction or multiple inputs combining smaller amounts, and at most two outputs: one for the payment, and one returning the change, if any, back to the sender.



It should be noted that fan-out, where a transaction depends on several transactions, and those transactions depend on many more, is not a problem here. There is never the need to extract a complete standalone copy of a transaction's history.

10. Privacy

The traditional banking model achieves a level of privacy by limiting access to information to the parties involved and the trusted third party. The necessity to announce all transactions publicly precludes this method, but privacy can still be maintained by breaking the flow of information in another place: by keeping public keys anonymous. The public can see that someone is sending an amount to someone else, but without information linking the transaction to anyone. This is similar to the level of information released by stock exchanges, where the time and size of individual trades, the "tape", is made public, but without telling who the parties were.



As an additional firewall, a new key pair should be used for each transaction to keep them from being linked to a common owner. Some linking is still unavoidable with multi-input transactions, which necessarily reveal that their inputs were owned by the same owner. The risk is that if the owner of a key is revealed, linking could reveal other transactions that belonged to the same owner.

11. Calculations

We consider the scenario of an attacker trying to generate an alternate chain faster than the honest chain. Even if this is accomplished, it does not throw the system open to arbitrary changes, such as creating value out of thin air or taking money that never belonged to the attacker. Nodes are not going to accept an invalid transaction as payment, and honest nodes will never accept a block containing them. An attacker can only try to change one of his own transactions to take back money he recently spent.

The race between the honest chain and an attacker chain can be characterized as a Binomial Random Walk. The success event is the honest chain being extended by one block, increasing its lead by +1, and the failure event is the attacker's chain being extended by one block, reducing the gap by -1.

The probability of an attacker catching up from a given deficit is analogous to a Gambler's Ruin problem. Suppose a gambler with unlimited credit starts at a deficit and plays potentially an infinite number of trials to try to reach breakeven. We can calculate the probability he ever reaches breakeven, or that an attacker ever catches up with the honest chain, as follows [8]:

p = probability an honest node finds the next block

q = probability the attacker finds the next block

q_z = probability the attacker will ever catch up from z blocks behind

$$q_z = \begin{cases} 1 & \text{if } p \leq q \\ (q/p)^z & \text{if } p > q \end{cases}$$

Given our assumption that $p > q$, the probability drops exponentially as the number of blocks the attacker has to catch up with increases. With the odds against him, if he doesn't make a lucky lunge forward early on, his chances become vanishingly small as he falls further behind.

We now consider how long the recipient of a new transaction needs to wait before being sufficiently certain the sender can't change the transaction. We assume the sender is an attacker who wants to make the recipient believe he paid him for a while, then switch it to pay back to himself after some time has passed. The receiver will be alerted when that happens, but the sender hopes it will be too late.

The receiver generates a new key pair and gives the public key to the sender shortly before signing. This prevents the sender from preparing a chain of blocks ahead of time by working on it continuously until he is lucky enough to get far enough ahead, then executing the transaction at that moment. Once the transaction is sent, the dishonest sender starts working in secret on a parallel chain containing an alternate version of his transaction.

The recipient waits until the transaction has been added to a block and z blocks have been linked after it. He doesn't know the exact amount of progress the attacker has made, but assuming the honest blocks took the average expected time per block, the attacker's potential progress will be a Poisson distribution with expected value:

$$\lambda = z \frac{q}{p}$$

To get the probability the attacker could still catch up now, we multiply the Poisson density for each amount of progress he could have made by the probability he could catch up from that point:

$$\sum_{k=0}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} \begin{cases} (q/p)^{(z-k)} & \text{if } k \leq z \\ 1 & \text{if } k > z \end{cases}$$

Rearranging to avoid summing the infinite tail of the distribution...

$$1 - \sum_{k=0}^z \frac{\lambda^k e^{-\lambda}}{k!} (1 - (q/p)^{(z-k)})$$

Converting to C code...

```
#include <math.h>
double AttackerSuccessProbability(double q, int z)
{
    double p = 1.0 - q;
    double lambda = z * (q / p);
    double sum = 1.0;
    int i, k;
    for (k = 0; k <= z; k++)
    {
        double poisson = exp(-lambda);
        for (i = 1; i <= k; i++)
            poisson *= lambda / i;
        sum -= poisson * (1 - pow(q / p, z - k));
    }
    return sum;
}
```

Running some results, we can see the probability drop off exponentially with z .

```

q=0.1
z=0    P=1.000000
z=1    P=0.2045873
z=2    P=0.0509779
z=3    P=0.0131722
z=4    P=0.0034552
z=5    P=0.0009137
z=6    P=0.0002428
z=7    P=0.0000647
z=8    P=0.0000173
z=9    P=0.0000046
z=10   P=0.0000012

q=0.3
z=0    P=1.000000
z=5    P=0.1773523
z=10   P=0.0416605
z=15   P=0.0101008
z=20   P=0.0024804
z=25   P=0.0006132
z=30   P=0.0001522
z=35   P=0.0000379
z=40   P=0.0000095
z=45   P=0.0000024
z=50   P=0.0000006

```

Solving for P less than 0.1%...

```

P < 0.001
q=0.10  z=5
q=0.15  z=8
q=0.20  z=11
q=0.25  z=15
q=0.30  z=24
q=0.35  z=41
q=0.40  z=89
q=0.45  z=340

```

12. Conclusion

We have proposed a system for electronic transactions without relying on trust. We started with the usual framework of coins made from digital signatures, which provides strong control of ownership, but is incomplete without a way to prevent double-spending. To solve this, we proposed a peer-to-peer network using proof-of-work to record a public history of transactions that quickly becomes computationally impractical for an attacker to change if honest nodes control a majority of CPU power. The network is robust in its unstructured simplicity. Nodes work all at once with little coordination. They do not need to be identified, since messages are not routed to any particular place and only need to be delivered on a best effort basis. Nodes can leave and rejoin the network at will, accepting the proof-of-work chain as proof of what happened while they were gone. They vote with their CPU power, expressing their acceptance of valid blocks by working on extending them and rejecting invalid blocks by refusing to work on them. Any needed rules and incentives can be enforced with this consensus mechanism.

References

- [1] W. Dai, "b-money," <http://www.weidai.com/bmoney.txt>, 1998.
- [2] H. Massias, X.S. Avila, and J.-J. Quisquater, "Design of a secure timestamping service with minimal trust requirements," In *20th Symposium on Information Theory in the Benelux*, May 1999.
- [3] S. Haber, W.S. Stornetta, "How to time-stamp a digital document," In *Journal of Cryptology*, vol 3, no 2, pages 99-111, 1991.
- [4] D. Bayer, S. Haber, W.S. Stornetta, "Improving the efficiency and reliability of digital time-stamping," In *Sequences II: Methods in Communication, Security and Computer Science*, pages 329-334, 1993.
- [5] S. Haber, W.S. Stornetta, "Secure names for bit-strings," In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, pages 28-35, April 1997.
- [6] A. Back, "Hashcash - a denial of service counter-measure," <http://www.hashcash.org/papers/hashcash.pdf>, 2002.
- [7] R.C. Merkle, "Protocols for public key cryptosystems," In *Proc. 1980 Symposium on Security and Privacy*, IEEE Computer Society, pages 122-133, April 1980.
- [8] W. Feller, "An introduction to probability theory and its applications," 1957.



BITCOINGOLD

Bitcoin Gold (BTG)

www.btcgpu.org
press@btcgpu.org
support@btcgpu.org

Abstract. Bitcoin Gold is a community-led project to create an experimental hard fork of Bitcoin to a new proof-of-work algorithm. The purpose for doing this is to make Bitcoin mining decentralized again. Satoshi Nakamoto's idealistic vision of "one CPU one vote" has been superseded by a reality where the manufacture and distribution of mining equipment has become dominated by a very small number of entities, some of whom have engaged in abusive practices against individual miners and the Bitcoin network as a whole. Bitcoin Gold will provide an opportunity for countless new people around the world to participate in the mining process with widely-available consumer hardware that is manufactured and distributed by reputable mainstream corporations. A more decentralized, democratic mining infrastructure is more resilient and more in line with Satoshi's original vision. Perhaps, if the Bitcoin Gold experiment is judged by the community to be a success, it may one day help build consensus for a proof-of-work hard fork on Bitcoin itself.

Introduction

Bitcoin was created for many different reasons and every day, people find new reasons to adopt Bitcoin. One of the historical reason is that people do not trust states or banks or any such intermediaries to control their money.

One of the central component of the Bitcoin architecture is mining. Simply put miners verify every transaction and compete with each other to get rewards. To get the reward, a miner has to solve a math problem before anyone else in the network.

Back in the days, a miner would be any geek with a computer, willing to trade electricity for Bitcoins. Today, a miner is usually a huge warehouse full of very advanced computers, constantly running to solve the math problems as fast as possible.



BITCOINGOLD

As it becomes more and more difficult to mine Bitcoin, more capital is required to operate profitable mining operations. They often are located in a country where the electricity is very cheap. Today, a great majority of the miners are located in China because they have access to cheap electricity.

In Satoshi Nakamoto's white paper, one of the main idea was that every CPU was going to be an equally important part of the network. We want Bitcoin to be a shared and independent currency. We don't want any fat cat to drive our monetary architecture.

The importance of miners in the network is constantly growing. To preserve the independence of the Bitcoin ecosystem from miners' influence, some people thought that it would be a good idea to change the bitcoin protocol in such a way that more people can have access to Bitcoin mining.

That's why Bitcoin Gold was born, in order to bring Bitcoin mining back to the "people".

Origins of Bitcoin Gold

In July 2017, Jack Liao, CEO of LightingAsic and BitExchange, made an announcement that he was working on a hard fork of Bitcoin to change the proof-of-work algorithm from the SHA256 algorithm originally selected by Satoshi Nakamoto to Equihash. The effect of this change will be to enable a whole new class of individuals and businesses to participate in mining this new branch of the Bitcoin blockchain without being required to purchase specialized equipment that is primarily manufactured by one firm that competes against its own customers with newer, more efficient versions of the old equipment that it sells at a high markup.

Given the dysfunctional current reality of the Bitcoin mining sector, it is no wonder that there is a tremendous appetite for a proof-of-work change hard fork. Since the Bitcoin Gold project was announced, it has grown rapidly, attracting developers, miners, and supporters from across the globe.



BITCOINGOLD

Mechanics of a Hard Fork

Bitcoin is a distributed consensus system. All Bitcoin full nodes are running software that enforces the same consensus rules; full nodes that enforce different consensus rules are not part of the Bitcoin network, by definition. If a miner finds a new block that follows the network consensus rules and broadcasts it to the network, all full nodes in the network will accept that block and all of the transactions in it as valid, and miners will build the next block on top of that one. A blockchain hard fork occurs when a block is mined that does not comply with the network consensus rules.

Prior to BTC block 478558, Bitcoin nodes and Bitcoin Cash nodes were still enforcing the same consensus rules and accepting the same blockchain as valid. But from that block onward, Bitcoin Cash's new consensus rules came into effect, which caused Bitcoin nodes to reject blocks that were mined by miners using Bitcoin Cash software, and Bitcoin Cash nodes to reject blocks that were mined by miners who continued to mine with Bitcoin software. Thus, the network bifurcated.

The Bitcoin blockchain continued to add a new block every 10 minutes on average, but Bitcoin Cash began building a new blockchain that branched away from Bitcoin. This had the effect of creating a new cryptocurrency that shares the same transaction history and ownership distribution up until the fork block, but then diverges from it.

Bitcoin Gold changes different consensus rules than Bitcoin Cash did, but it will fork from Bitcoin in the same manner - by enforcing new consensus rules as of a predetermined BTC block height. The new rules will come into effect at block 491407. From this block onward, Bitcoin Gold miners will begin building a new branch of the Bitcoin blockchain. This new branch is a cryptocurrency with same transaction history and ownership distribution as Bitcoin at the fork block; if you hold BTC, you will automatically receive an equal amount of BTG.



BTC/GOLD

Here are some of the differences between Bitcoin Gold and other forks of Bitcoin:

Comparison BTC/BTG/BCH/B2X	BITCOIN BTC	BITCOIN GOLD BTG	BITCOIN CASH BCH	SEGWIT 2X B2X
Supply	21 Million	21 Million	21 Million	21 Million
PoW algorithm	SHA256	Equihash	SHA256	SHA256
Mining Hardware	ASIC	GPU	ASIC	ASIC
Block Interval	10 Minutes	10 Minutes	10 Minutes	10 Minutes
Block size (actual)	1M (2-4M)	1M (2-4M)	8M (8M)	2M (4-8M)
Difficulty adjustment	2 Weeks	Every block	2 Weeks + EDA	2 Weeks
Segwit	✓	✓	✗	✓
Replay protection	●	✓	✓	✗
Unique address format	●	✓	✗	✗

Proof-of-Work Algorithm

Bitcoin mining is a proof-of-work system that implements “a distributed timestamp server on a peer-to-peer basis.” This is how the Bitcoin manages to maintain consensus across a vast, globally-distributed, permissionless network of nodes.

Satoshi Nakamoto chose SHA256 as the algorithm to use in the original design of Bitcoin’s PoW system. SHA256 served Bitcoin well during the early years of its existence, but as Bitcoin became more popular and more valuable, competition in mining became more fierce. Skilled engineers from a small number of companies developed Application Specific Integrated Circuits (ASICs) that could perform SHA256 calculations millions of times faster and more efficiently than any other computer. This made non-specialized computer hardware obsolete for mining Bitcoin. Satoshi’s vision of “one-CPU-one-vote” was replaced by one-ASIC-one-vote.



BITCOINGOLD

Now, the only way to participate in Bitcoin mining is to buy hardware from one of those manufacturers - the biggest of which is believed to manufacture over 70% of the global supply of SHA256 ASICs. This has led to a situation where one entity can hold the entire network hostage, and this is exactly what happened when the backwards compatible Segregated Witness upgrade was blocked by a faction of miners, despite there being universal consensus from Bitcoin experts that it should be activated.

In order to counteract this concentration of power in the mining sector, Bitcoin Gold will implement a new proof-of-work algorithm - Equihash. Replacing the SHA256 algorithm means that all of the ASICs designed for Bitcoin will be useless for mining Bitcoin Gold. Equihash is a memory-hard algorithm that can be most efficiently solved by GPUs - a standard type of computer and smartphone hardware that is manufactured by mainstream companies and available around the world. With ASIC manufacturers out of the picture, Bitcoin Gold will provide an opportunity for a whole new class entrepreneurs and investors to get involved with mining. Bitcoin Gold mining will be decentralized again, closer to Satoshi's original vision.

ASIC-resistance is a permanent attribute of Bitcoin Gold. It is much more difficult to create ASICs for a memory hard algorithm like Equihash than SHA256, however it is not impossible. If the day ever comes when Equihash ASICs begin to proliferate and mining begins to centralize again, Bitcoin Gold will have another hard fork to implement a new PoW algorithm.

Difficulty Adjustment Algorithm

In Bitcoin, the difficulty of mining adjusts every 2016 blocks (approximately two weeks) in order to maintain an average interval of 10 minutes between blocks. If the average time between blocks was less than 10 minutes, the difficulty will increase; if the average time was more than 10 minutes, the difficulty will decrease.

Bitcoin Gold will adopt a difficulty adjustment algorithm called DigiShield V3. The idea behind it is to look at how much time has elapsed between the most recent block and the median of a set number of preceding blocks, and to adjust the difficulty every block to target a 10 minute block interval. This more responsive difficulty adjustment algorithm is extremely useful in protecting against big swings in the total amount of hash power. Such swings can result in extreme deviation from the normal 10 minute target block interval. Bitcoin Cash attempted to protect against this risk by implementing an "emergency difficulty adjustment" algorithm, but that had the catastrophic effect of causing sometimes 50 blocks to be mined in one hour, and other times more than 12 hours between two blocks.



BITCOINGOLD

Replay Protection

The risk of a replay attack is inherent to every cryptocurrency hard fork and has to be taken into consideration to protect users from losing their funds. A hard fork is an exact duplicate of the blockchain, and as such, a transaction that is broadcast publicly to the network can be replayed on both sides of a fork, unless replay protection is implemented.

Bitcoin Gold will implement a solution called *SIGHASH_FORK_ID* replay protection. It is an effective two-way replay protection mechanism that enforces a new algorithm to calculate the hash of a transaction so that all the new Bitcoin transactions will be invalid in Bitcoin Gold blockchain and vice versa. Bitcoin Gold will implement replay protection BEFORE THE LAUNCH.

Unique Address Format

By default, both sides of a cryptocurrency hard fork will continue to use the same address format. That means it's possible to send coins to an address on the other blockchain unintentionally, which can cause users to lose funds by mistake. Bitcoin Cash, for example, is a hard fork that did not change the address format; its addresses are indistinguishable from Bitcoin addresses. There have been many reports of people accidentally sending their BTC to a BCC address and vice versa. In some cases these coins could be permanently lost.

In order to ensure that this potential confusion does not exist in Bitcoin Gold, a unique address format will be implemented. The prefix of PUBKEY_ADDRESS and SCRIPT_ADDRESS will be changed to a new prefix (yet to be determined) that can easily be distinguished from Bitcoin addresses.

How to Acquire Bitcoin Gold

The hardfork will occur on block 491407. To acquire free Bitcoin Gold you simply have to hold Bitcoin at the time of the fork. If you hold BTC at that time, you will automatically receive an equal amount of BTG at the same address (new and old address format are convertible), spendable with the same private keys, when the Bitcoin Gold network launches in November. It is also very important to make a backup of your private key and/or keep the [mnemonic phrase](#) required to recover your wallet.



BITCOINGOLD

However, if you have your BTC on an exchange or custodial service without access to the private key, then you have to make sure that the service will support Bitcoin Gold after the fork. If you have any doubts about that, then you would be advised to transfer your BTC to one of the many reputable services that will support it.

Timeline

Step 1: The hard fork occurs: a ‘snapshot’ of the blockchain is taken

Usually a hard fork will happen at the same time when Bitcoin reaches the fork block. However, Bitcoin Gold uses a different way to launch the hard fork: by “taking a snapshot” of the Bitcoin blockchain before the fork block height 491407. Instead of forking immediately, the Bitcoin Gold p2p network will launch a few days later from that snapshot.

When Bitcoin reaches the block 491407, nothing special will happen. Bitcoin block 491407 will be mined with SHA256 as normal. No block will be mined in the Bitcoin Gold p2p network because it is not launched yet.

However, when the full node client of Bitcoin Gold is ready a few days later, instead of mining from the latest Bitcoin block, Bitcoin Gold will start to mine its own 491407th block on top of block 491406. Bitcoin Gold full nodes will only accept a block 491407 that is mined with Equihash, so they will not recognize BTC block 491407 as a valid BTG block.

At the same time, Bitcoin already have a longer blockchain. That's why it's called a “snapshot hard fork”. We didn't follow the common realtime hard fork pattern because a PoW change means there will always be a gap between the fork block.

The first Equihash block will be block 491407 of the Bitcoin Gold blockchain, and from that point on GPU miners participating in the Bitcoin Gold network will begin mining more Equihash blocks on top of it. In this way, the Bitcoin blockchain will bifurcate and a new coin - Bitcoin Gold (BTG) - will be created. Everyone who holds BTC at block 491406 will then control an equal amount of coins on the BTG blockchain branch, which can be spent at any time in the future with the corresponding private keys.



BITCOINGOLD

Step 2: The BTG blockchain is activated

If you have BTC in a paper wallet, hardware wallet, multi-signature address, or any other form of secure private key storage, you will be able to spend your corresponding BTG at any time in the future. There is no expiration date for your BTG. If you have BTC in cold storage that you did not plan to touch for many years, do not change your plans because of this fork. Your BTG will still be there decades from now.

In 491407 hard fork is the one and only opportunity to get initial BTG. After that time, your options to acquire it will be to buy it on an exchange like any other cryptocurrency, to mine it with your own computer hardware (GPUs), or to earn it by trading your goods and services for it.

Cryptocurrency exchanges are custodial businesses, which means they control your private keys, not you. When the Bitcoin Gold fork occurs on block height 491407, any exchange that is holding BTC on your behalf will also receive the corresponding BTG. While they should credit your account with the equal amount of BTG, there is no legal authority that can force them to do so. The Bitcoin Gold home page will display the names and logos of exchanges that have promised to credit their users with BTG at the 1:1 ratio. If your exchange is not shown, please consider transferring your BTC to a supporting exchange or withdraw to a personal wallet where you control the private keys.



BITCOINGOLD

Financial Strategy

In order to support the current and future development of Bitcoin Gold, the first blocks after the fork will have a reduced difficulty level that will allow the development team to mine these blocks very rapidly, and then the new difficulty adjustment algorithm will kick in and everyone will have the opportunity to mine on equal footing. As a result, the Bitcoin Gold development team will manage 0.476% of the total coin supply, which will be the main source of funding for all future development of this project, including valuable research and testing that may one day help bring about consensus for a proof-of-work change on Bitcoin itself.

- The initial BTG mined by the Bitcoin Gold (0.476%) development team will be held in multi-signature wallets.
- 60% of the funds will be time-locked and released in proportional amounts over the course of three years to cover the development costs.
- All significant expenditures will be made fully transparent according to the best practices of similar open source projects.
- The majority of funds will be allocated as developer bounties, which will be published as issues in the [BTCPGPU](#) GitHub repository.
- Everyone is able to participate in the Bitcoin Gold developer bounty program; to win the bounty, you must provide the open source code that meets the specific requirements.

The Bitcoin community will be able to support these bounties by buying or holding BTG, as the price of the coin will determine how strong of an incentive these bounties are, and how soon these features can be created. Keep in mind that most of these development bounties are designed to benefit the entire Bitcoin ecosystem, not only the Bitcoin Gold fork. Bitcoin Gold itself was designed to be a feature of Bitcoin, not a rival.

Some of these essential functions will be performed by full-time employees while others will be outsourced to third-party professional services. All of these expenditures will be made as transparent as possible without compromising operational security.



BITCOINGOLD

BITCOIN GOLD 40% BTG

Startup expenses

Bounties and app collaboration	7%
Pre-fork costs	5%
Community development	3%
Initial reward for core team	5%
Yearly expenses	20%
Total	40%

BITCOIN GOLD 60% BTG

Time-locked funds; 20% released per year

Development	30%
Ecosystem	15%
Community	15%
Total	60%



BITCOINGOLD

Future development:

- Core protocol
- Lightning network
- Bech32 addresses
- Sidechains
- Cross-chain atomic swaps
- Decentralized exchange

Operational and infrastructure costs:

- Servers:
 - 12+ full nodes on 6 continents
 - 5+ DNS seeds
 - Website
- Domain fee
- System administration
- Security and penetration testing by third-party

Future social action:

- Economic Development Fund:
 - BTG debit card program (Latin America)
 - Decentralized fiat-crypto brokerage network (Global)
- Blockchain Education Fund:
 - Investment in the content creators and influencers who most effectively contribute to rising Bitcoin awareness and adoption.
- GPU Mining Infrastructure Fund:
 - Small/mid-scale individual/business loans for GPU mining hardware operations.
 - Developer bounties for user-friendly mining applications that can bring mining to a non-technical, multi-lingual audience.

Future communication costs:

- Meetups and developer conferences
- Social media
- Design assets
- Press releases



BITCOINGOLD

Conclusion

Bitcoin Gold is a free open source project that was created by a small group of Bitcoin enthusiasts from diverse backgrounds. In contrast to the other prominent Bitcoin forks, Bitcoin Gold was specifically designed from the beginning to inspire innovation in the Bitcoin ecosystem and give value to the vision of decentralization. Whereas the others were born from hostility and an ambition to dominate, Bitcoin Gold arises from a desire to protect Bitcoin and ensure that it not only maintains its position as the dominant cryptocurrency but continues to grow until its liberating roots stretch deep into the economic life of all nations.

Cosmos

A Network of Distributed Ledgers

Jae Kwon jae@tendermint.com

Ethan Buchman ethan@tendermint.com

For discussions, join our [community chat](#)!

NOTE: If you can read this on GitHub, then we're still actively developing this document. Please check regularly for updates!

Introduction

The combined success of the open-source ecosystem, decentralized file-sharing, and public cryptocurrencies has inspired an understanding that decentralized internet protocols can be used to radically improve socio-economic infrastructure. We have seen specialized blockchain applications like Bitcoin [1] (a cryptocurrency), Zerocash [2] (a cryptocurrency for privacy), and generalized smart contract platforms such as Ethereum [3], with countless distributed applications for the Etherium Virtual Machine (EVM) such as Augur (a prediction market) and TheDAO [4] (an investment club).

To date, however, these blockchains have suffered from a number of drawbacks, including their gross energy inefficiency, poor or limited performance, and immature governance mechanisms. Proposals to scale Bitcoin's transaction throughput, such as Segregated-Witness [5] and BitcoinNG [6], are vertical scaling solutions that remain limited by the capacity of a single physical machine, in order to ensure the property of complete auditability. The Lightning Network [7] can help scale Bitcoin transaction

volume by leaving some transactions off the ledger completely, and is well suited for micropayments and privacy-preserving payment rails, but may not be suitable for more generalized scaling needs.

An ideal solution is one that allows multiple parallel blockchains to interoperate while retaining their security properties. This has proven difficult, if not impossible, with proof-of-work. Merged mining, for instance, allows the work done to secure a parent chain to be reused on a child chain, but transactions must still be validated, in order, by each node, and a merge-mined blockchain is vulnerable to attack if a majority of the hashing power on the parent is not actively merge-mining the child. An academic review of [alternative blockchain network architectures](#) is provided for additional context, and we provide summaries of other proposals and their drawbacks in [Related Work](#).

Here we present Cosmos, a novel blockchain network architecture that addresses all of these problems. Cosmos is a network of many independent blockchains, called zones. The zones are powered by Tendermint Core [8], which provides a high-performance, consistent, secure [PBFT-like](#) consensus engine, where strict [fork-accountability](#) guarantees hold over the behaviour of malicious actors. Tendermint Core's BFT consensus algorithm is well suited for scaling public proof-of-stake blockchains.

The first zone on Cosmos is called the Cosmos Hub. The Cosmos Hub is a multi-asset proof-of-stake cryptocurrency with a simple governance mechanism which enables the network to adapt and upgrade. In addition, the Cosmos Hub can be extended by connecting other zones.

The hub and zones of the Cosmos network communicate with each other via an inter-blockchain communication (IBC) protocol, a kind of virtual UDP or TCP for blockchains. Tokens can be transferred from one zone to another securely and quickly

without the need for exchange liquidity between zones. Instead, all inter-zone token transfers go through the Cosmos Hub, which keeps track of the total amount of tokens held by each zone. The hub isolates each zone from the failure of other zones. Because anyone can connect a new zone to the Cosmos Hub, zones allow for future-compatibility with new blockchain innovations.

Tendermint

In this section we describe the Tendermint consensus protocol and the interface used to build applications with it. For more details, see the [appendix](#).

Validators

In classical Byzantine fault-tolerant (BFT) algorithms, each node has the same weight. In Tendermint, nodes have a non-negative amount of *voting power*, and nodes that have positive voting power are called *validators*. Validators participate in the consensus protocol by broadcasting cryptographic signatures, or votes, to agree upon the next block.

Validators' voting powers are determined at genesis, or are changed deterministically by the blockchain, depending on the application. For example, in a proof-of-stake application such as the Cosmos Hub, the voting power may be determined by the amount of staking tokens bonded as collateral.

NOTE: Fractions like $\frac{2}{3}$ and $\frac{1}{3}$ refer to fractions of the total voting power, never the total number of validators, unless all the validators have equal weight. $>\frac{2}{3}$ means “more than $\frac{2}{3}$ ”, $\geq\frac{1}{3}$ means “at least $\frac{1}{3}$ ”.

Consensus

Tendermint is a partially synchronous BFT consensus protocol derived from the DLS consensus algorithm [20]. Tendermint is

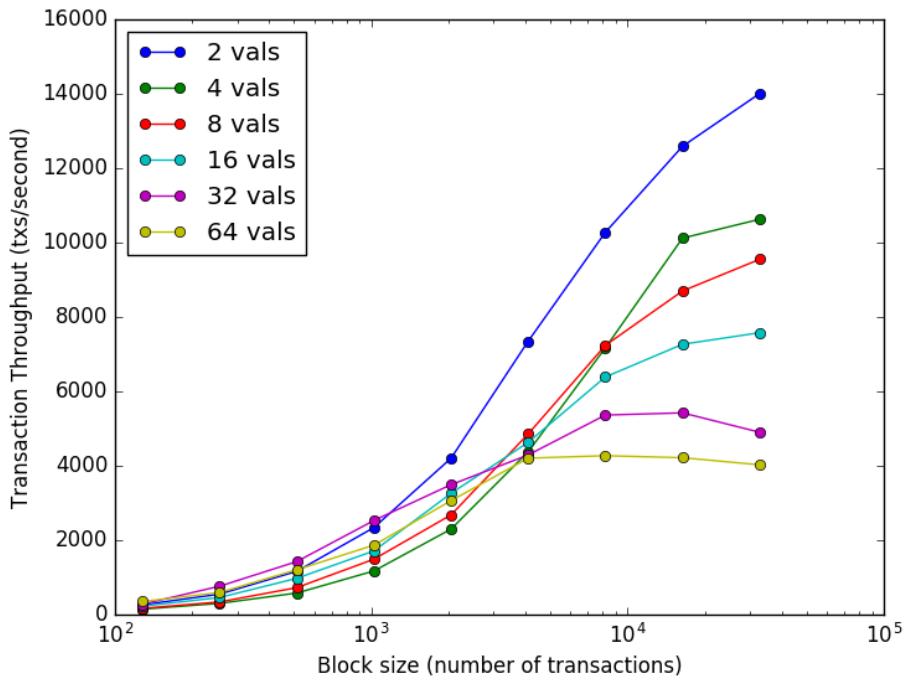
notable for its simplicity, performance, and [fork-accountability](#). The protocol requires a fixed known set of validators, where each validator is identified by their public key. Validators attempt to come to consensus on one block at a time, where a block is a list of transactions. Voting for consensus on a block proceeds in rounds. Each round has a round-leader, or proposer, who proposes a block. The validators then vote, in stages, on whether to accept the proposed block or move on to the next round. The proposer for a round is chosen deterministically from the ordered list of validators, in proportion to their voting power.

The full details of the protocol are described [here](#).

Tendermint's security derives from its use of optimal Byzantine fault-tolerance via super-majority ($>2/3$) voting and a locking mechanism. Together, they ensure that:

- $\geq 1/3$ voting power must be Byzantine to cause a violation of safety, where more than two values are committed.
- if any set of validators ever succeeds in violating safety, or even attempts to do so, they can be identified by the protocol. This includes both voting for conflicting blocks and broadcasting unjustified votes.

Despite its strong guarantees, Tendermint provides exceptional performance. In benchmarks of 64 nodes distributed across 7 datacenters on 5 continents, on commodity cloud instances, Tendermint consensus can process thousands of transactions per second, with commit latencies on the order of one to two seconds. Notably, performance of well over a thousand transactions per second is maintained even in harsh adversarial conditions, with validators crashing or broadcasting maliciously crafted votes. See the figure below for details.



Light Clients

A major benefit of Tendermint's consensus algorithm is simplified light client security, making it an ideal candidate for mobile and internet-of-things use cases. While a Bitcoin light client must sync chains of block headers and find the one with the most proof of work, Tendermint light clients need only to keep up with changes to the validator set, and then verify the $>2/3$ PreCommits in the latest block to determine the latest state.

Succinct light client proofs also enable [inter-blockchain communication](#).

Preventing Attacks

Tendermint has protective measures for preventing certain notable attacks, like [long-range-nothing-at-stake double spends](#) and [censorship](#). These are discussed more fully in the [appendix](#).

ABCI

The Tendermint consensus algorithm is implemented in a program called Tendermint Core. Tendermint Core is an application-agnostic “consensus engine” that can turn any deterministic blackbox application into a distributedly replicated blockchain. Tendermint Core connects to blockchain applications via the Application Blockchain Interface (ABCI) [17]. Thus, ABCI allows for blockchain applications to be programmed in any language, not just the programming language that the consensus engine is written in. Additionally, ABCI makes it possible to easily swap out the consensus layer of any existing blockchain stack.

We draw an analogy with the well-known cryptocurrency Bitcoin. Bitcoin is a cryptocurrency blockchain where each node maintains a fully audited Unspent Transaction Output (UTXO) database. If one wanted to create a Bitcoin-like system on top of ABCI, Tendermint Core would be responsible for

- Sharing blocks and transactions between nodes
- Establishing a canonical/imutable order of transactions (the blockchain)

Meanwhile, the ABCI application would be responsible for

- Maintaining the UTXO database
- Validating cryptographic signatures of transactions
- Preventing transactions from spending non-existent funds
- Allowing clients to query the UTXO database

Tendermint is able to decompose the blockchain design by offering a very simple API between the application process and consensus process.

Cosmos Overview

Cosmos is a network of independent parallel blockchains that are each powered by classical BFT consensus algorithms like Tendermint [1](#).

The first blockchain in this network will be the Cosmos Hub. The Cosmos Hub connects to many other blockchains (or zones) via a novel inter-blockchain communication protocol. The Cosmos Hub tracks numerous token types and keeps record of the total number of tokens in each connected zone. Tokens can be transferred from one zone to another securely and quickly without the need for a liquid exchange between zones, because all inter-zone coin transfers go through the Cosmos Hub.

This architecture solves many problems that the blockchain space faces today, such as application interoperability, scalability, and seamless upgradability. For example, zones derived from Bitcoind, Go-Ethereum, CryptoNote, ZCash, or any blockchain system can be plugged into the Cosmos Hub. These zones allow Cosmos to scale infinitely to meet global transaction demand. Zones are also a great fit for a distributed exchange, which will be supported as well.

Cosmos is not just a single distributed ledger, and the Cosmos Hub isn't a walled garden or the center of its universe. We are designing a protocol for an open network of distributed ledgers that can serve as a new foundation for future financial systems, based on principles of cryptography, sound economics, consensus theory, transparency, and accountability.

Tendermint-BFT

The Cosmos Hub is the first public blockchain in the Cosmos Network, powered by Tendermint's BFT consensus algorithm. The Tendermint open-source project was born in 2014 to address the speed, scalability, and environmental issues of Bitcoin's proof-of-work consensus algorithm. By using and improving upon proven

BFT algorithms developed at MIT in 1988 [20], the Tendermint team was the first to conceptually demonstrate a proof-of-stake cryptocurrency that addresses the nothing-at-stake problem suffered by first-generation proof-of-stake cryptocurrencies such as NXT and BitShares1.0.

Today, practically all Bitcoin mobile wallets use trusted servers to provide them with transaction verification. This is because proof-of-work requires waiting for many confirmations before a transaction can be considered irreversibly committed. Double-spend attacks have already been demonstrated on services like CoinBase.

Unlike other blockchain consensus systems, Tendermint offers instant and provably secure mobile-client payment verification. Since the Tendermint is designed to never fork at all, mobile wallets can receive instant transaction confirmation, which makes trustless and practical payments a reality on smartphones. This has significant ramifications for Internet of Things applications as well.

Validators in Cosmos have a similar role to Bitcoin miners, but instead use cryptographic signatures to vote. Validators are secure, dedicated machines that are responsible for committing blocks. Non-validators can delegate their staking tokens (called “atoms”) to any validator to earn a portion of block fees and atom rewards, but they incur the risk of getting punished (slashed) if the delegate validator gets hacked or violates the protocol. The proven safety guarantees of Tendermint BFT consensus, and the collateral deposit of stakeholders—validators and delegators—provide provable, quantifiable security for nodes and light clients.

Governance

Distributed public ledgers should have a constitution and a governance system. Bitcoin relies on the Bitcoin Foundation and

mining to coordinate upgrades, but this is a slow process.

Ethereum split into ETH and ETC after hard-forking to address TheDAO hack, largely because there was no prior social contract nor mechanism for making such decisions.

Validators and delegators on the Cosmos Hub can vote on proposals that can change preset parameters of the system automatically (such as the block gas limit), coordinate upgrades, as well as vote on amendments to the human-readable constitution that govern the policies of the Cosmos Hub. The constitution allows for cohesion among the stakeholders on issues such as theft and bugs (such as TheDAO incident), allowing for quicker and cleaner resolution.

Each zone can also have their own constitution and governance mechanism as well. For example, the Cosmos Hub could have a constitution that enforces immutability at the Hub (no roll-backs, save for bugs of the Cosmos Hub node implementation), while each zone can set their own policies regarding roll-backs.

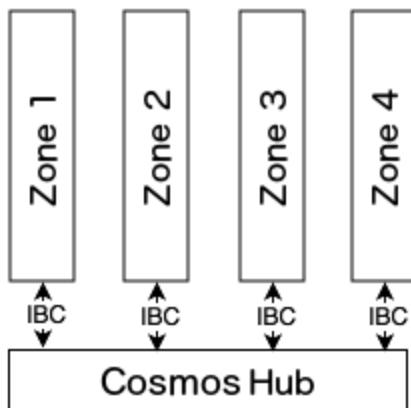
By enabling interoperability among differing policy zones, the Cosmos network gives its users ultimate freedom and potential for permissionless experimentation.

The Hub and Zones

Here we describe a novel model of decentralization and scalability. Cosmos is a network of many blockchains powered by Tendermint. While existing proposals aim to create a “single blockchain” with total global transaction ordering, Cosmos permits many blockchains to run concurrently with one another while retaining interoperability.

At the basis, the Cosmos Hub manages many independent blockchains called “zones” (sometimes referred to as “shards”, in reference to the database scaling technique known as “sharding”).

A constant stream of recent block commits from zones posted on the Hub allows the Hub to keep up with the state of each zone. Likewise, each zone keeps up with the state of the Hub (but zones do not keep up with each other except indirectly through the Hub). Packets of information are then communicated from one zone to another by posting Merkle-proofs as evidence that the information was sent and received. This mechanism is called inter-blockchain communication, or IBC for short.



Any of the zones can themselves be hubs to form an acyclic graph, but for the sake of clarity we will only describe the simple configuration where there is only one hub, and many non-hub zones.

The Hub

The Cosmos Hub is a blockchain that hosts a multi-asset distributed ledger, where tokens can be held by individual users or by zones themselves. These tokens can be moved from one zone to another in a special IBC packet called a "coin packet". The hub is responsible for preserving the global invariance of the total amount of each token across the zones. IBC coin packet transactions must be committed by the sender, hub, and receiver blockchains.

Since the Cosmos Hub acts as the central ledger for the whole system, the security of the Hub is of paramount importance. While each zone may be a Tendermint blockchain that is secured by as few as 4 (or even less if BFT consensus is not needed), the Hub must be secured by a globally decentralized set of validators that can withstand the most severe attack scenarios, such as a continental network partition or a nation-state sponsored attack.

The Zones

A Cosmos zone is an independent blockchain that exchanges IBC messages with the Hub. From the Hub's perspective, a zone is a multi-asset dynamic-membership multi-signature account that can send and receive tokens using IBC packets. Like a cryptocurrency account, a zone cannot transfer more tokens than it has, but can receive tokens from others who have them. A zone may be designated as an "source" of one or more token types, granting it the power to inflate that token supply.

Atoms of the Cosmos Hub may be staked by validators of a zone connected to the Hub. While double-spend attacks on these zones would result in the slashing of atoms with Tendermint's fork-accountability, a zone where $>2/3$ of the voting power are Byzantine can commit invalid state. The Cosmos Hub does not verify or execute transactions committed on other zones, so it is the responsibility of users to send tokens to zones that they trust. In the future, the Cosmos Hub's governance system may pass Hub improvement proposals that account for zone failures. For example, outbound token transfers from some (or all) zones may be throttled to allow for the emergency circuit-breaking of zones (a temporary halt of token transfers) when an attack is detected.

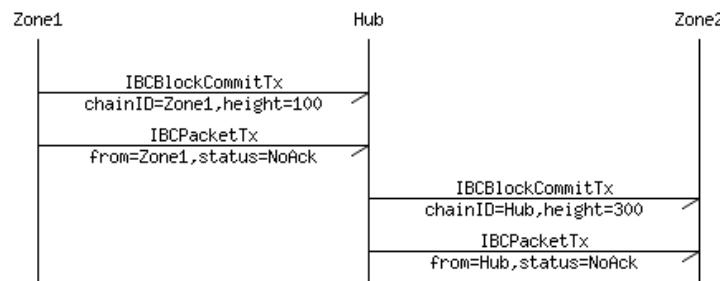
Inter-blockchain Communication (IBC)

Now we look at how the Hub and zones communicate with each other. For example, if there are three blockchains, "Zone1", "Zone2",

and “Hub”, and we wish for “Zone1” to produce a packet destined for “Zone2” going through “Hub”. To move a packet from one blockchain to another, a proof is posted on the receiving chain. The proof states that the sending chain published a packet for the alleged destination. For the receiving chain to check this proof, it must be able keep up with the sender’s block headers. This mechanism is similar to that used by sidechains, which requires two interacting chains to be aware of one another via a bidirectional stream of proof-of-existence datagrams (transactions).

The IBC protocol can naturally be defined using two types of transactions: an **IBCBLOCKCOMMITTX** transaction, which allows a blockchain to prove to any observer of its most recent block-hash, and an **IBCPACKETTX** transaction, which allows a blockchain to prove to any observer that the given packet was indeed published by the sender’s application, via a Merkle-proof to the recent block-hash.

By splitting the IBC mechanics into two separate transactions, we allow the native fee market-mechanism of the receiving chain to determine which packets get committed (i.e. acknowledged), while allowing for complete freedom on the sending chain as to how many outbound packets are allowed.



In the example above, in order to update the block-hash of “Zone1” on “Hub” (or of “Hub” on “Zone2”), an **IBCBLOCKCOMMITTX**

transaction must be posted on “Hub” with the block-hash of “Zone1” (or on “Zone2” with the block-hash of “Hub”).

See [IBCBlockCommitTx](#) and [IBCPacketTx](#) for more information on the two IBC transaction types.

Use Cases

Distributed Exchange

In the same way that Bitcoin is more secure by being a distributed, mass-replicated ledger, we can make exchanges less vulnerable to external and internal hacks by running it on the blockchain. We call this a distributed exchange.

What the cryptocurrency community calls a decentralized exchange today are based on something called “atomic cross-chain” (AXC) transactions. With an AXC transaction, two users on two different chains can make two transfer transactions that are committed together on both ledgers, or none at all (i.e. atomically). For example, two users can trade bitcoins for ether (or any two tokens on two different ledgers) using AXC transactions, even though Bitcoin and Ethereum are not connected to each other. The benefit of running an exchange on AXC transactions is that neither users need to trust each other or the trade-matching service. The downside is that both parties need to be online for the trade to occur.

Another type of decentralized exchange is a mass-replicated distributed exchange that runs on its own blockchain. Users on this kind of exchange can submit a limit order and turn their computer off, and the trade can execute without the user being online. The blockchain matches and completes the trade on behalf of the trader.

A centralized exchange can create a deep orderbook of limit orders and thereby attract more traders. Liquidity begets more liquidity in the exchange world, and so there is a strong network effect (or at least a winner-take-most effect) in the exchange business. The current leader for cryptocurrency exchanges today is Poloniex with a 24-hour volume of \$20M, and in second place is Bitfinex with a 24-hour volume of \$5M. Given such strong network effects, it is unlikely for AXC-based decentralized exchanges to win volume over the centralized exchanges. For a decentralized exchange to compete with a centralized exchange, it would need to support deep orderbooks with limit orders. Only a distributed exchange on a blockchain can provide that.

Tendermint provides additional benefits of faster transaction commits. By prioritizing fast finality without sacrificing consistency, zones in Cosmos can finalize transactions fast – for both exchange order transactions as well as IBC token transfers to and from other zones.

Given the state of cryptocurrency exchanges today, a great application for Cosmos is the distributed exchange (aka the Cosmos DEX). The transaction throughput capacity as well as commit latency can be comparable to those of centralized exchanges. Traders can submit limit orders that can be executed without both parties having to be online. And with Tendermint, the Cosmos hub, and IBC, traders can move funds in and out of the exchange to and from other zones with speed.

Bridging to Other Cryptocurrencies

A privileged zone can act as the source of a bridged token of another cryptocurrency. A bridge is similar to the relationship between a Cosmos hub and zone; both must keep up with the latest blocks of the other in order to verify proofs that tokens have moved from one to the other. A "bridge-zone" on the Cosmos network keeps up with the Hub as well as the other

cryptocurrency. The indirection through the bridge-zone allows the logic of the Hub to remain simple and agnostic to other blockchain consensus strategies such as Bitcoin's proof-of-work mining.

Sending Tokens to the Cosmos Hub

Each bridge-zone validator would run a Tendermint-powered blockchain with a special ABCI bridge-app, but also a full-node of the “origin” blockchain.

When new blocks are mined on the origin, the bridge-zone validators will come to agreement on committed blocks by signing and sharing their respective local view of the origin’s blockchain tip. When a bridge-zone receives payment on the origin (and sufficient confirmations were agreed to have been seen in the case of a PoW chain such as Ethereum or Bitcoin), a corresponding account is created on the bridge-zone with that balance.

In the case of Ethereum, the bridge-zone can share the same validator-set as the Cosmos Hub. On the Ethereum side (the origin), a bridge-contract would allow ether holders to send ether to the bridge-zone by sending it to the bridge-contract on Ethereum. Once ether is received by the bridge-contract, the ether cannot be withdrawn unless an appropriate IBC packet is received by the bridge-contract from the bridge-zone. The bridge-contract tracks the validator-set of the bridge-zone, which may be identical to the Cosmos Hub’s validator-set.

In the case of Bitcoin, the concept is similar except that instead of a single bridge-contract, each UTXO would be controlled by a threshold multisignature P2SH subscript. Due to the limitations of the P2SH system, the signers cannot be identical to the Cosmos Hub validator-set.

Withdrawing Tokens from Cosmos Hub

Ether on the bridge-zone (“bridged-ether”) can be transferred to and from the Hub, and later be destroyed with a transaction that sends it to a particular withdrawal address on Ethereum. An IBC packet proving that the transaction occurred on the bridge-zone can be posted to the Ethereum bridge-contract to allow the ether to be withdrawn.

In the case of Bitcoin, the restricted scripting system makes it difficult to mirror the IBC coin-transfer mechanism. Each UTXO has its own independent pubscript, so every UTXO must be migrated to a new UTXO when there is a change in the set of Bitcoin escrow signers. One solution is to compress and decompress the UTXO-set as necessary to keep the total number of UTXOs down.

Total Accountability of Bridge Zones

The risk of such a bridgeging contract is a rogue validator set. $\geq \frac{1}{3}$ Byzantine voting power could cause a fork, withdrawing ether from the bridge-contract on Ethereum while keeping the bridged-ether on the bridge-zone. Worse, $> \frac{2}{3}$ Byzantine voting power can steal ether outright from those who sent it to the bridge-contract by deviating from the original bridgeging logic of the bridge-zone.

It is possible to address these issues by designing the bridge to be totally accountable. For example, all IBC packets, from the hub and the origin, might require acknowledgement by the bridge-zone in such a way that all state transitions of the bridge-zone can be efficiently challenged and verified by either the hub or the origin’s bridge-contract. The Hub and the origin should allow the bridge-zone validators to post collateral, and token transfers out of the bridge-contract should be delayed (and collateral unbonding period sufficiently long) to allow for any challenges to be made by independent auditors. We leave the design of the specification and implementation of this system open as a future Cosmos

improvement proposal, to be passed by the Cosmos Hub's governance system.

Ethereum Scaling

Solving the scaling problem is an open issue for Ethereum. Currently, Ethereum nodes process every single transaction and also store all the states. [link](#).

Since Tendermint can commit blocks much faster than Ethereum's proof-of-work, EVM zones powered by Tendermint consensus and operating on bridged-ether can provide higher performance to Ethereum blockchains. Additionally, though the Cosmos Hub and IBC packet mechanics does not allow for arbitrary contract logic execution per se, it can be used to coordinate token movements between Ethereum contracts running on different zones, providing a foundation for token-centric Ethereum scaling via sharding.

Multi-Application Integration

Cosmos zones run arbitrary application logic, which is defined at the beginning of the zone's life and can potentially be updated over time by governance. Such flexibility allows Cosmos zones to act as bridges to other cryptocurrencies such as Ethereum or Bitcoin, and it also permits derivatives of those blockchains, utilizing the same codebase but with a different validator set and initial distribution. This allows many existing cryptocurrency frameworks, such as those of Ethereum, Zerocash, Bitcoin, CryptoNote and so on, to be used with Tendermint Core, which is a higher performance consensus engine, on a common network, opening tremendous opportunity for interoperability across platforms. Furthermore, as a multi-asset blockchain, a single transaction may contain multiple inputs and outputs, where each input can be any token type, enabling Cosmos to serve directly as a platform for decentralized exchange, though orders are assumed

to be matched via other platforms. Alternatively, a zone can serve as a distributed fault-tolerant exchange (with orderbooks), which can be a strict improvement over existing centralized cryptocurrency exchanges which tend to get hacked over time.

Zones can also serve as blockchain-backed versions of enterprise and government systems, where pieces of a particular service that are traditionally run by an organization or group of organizations are instead run as a ABCI application on a certain zone, which allows it to inherit the security and interoperability of the public Cosmos network without sacrificing control over the underlying service. Thus, Cosmos may offer the best of both worlds for organizations looking to utilize blockchain technology but who are wary of relinquishing control completely to a distributed third party.

Network Partition Mitigation

Some claim that a major problem with consistency-favouring consensus algorithms like Tendermint is that any network partition which causes there to be no single partition with $>2/3$ voting power (e.g. $\geq 1/3$ going offline) will halt consensus altogether. The Cosmos architecture can help mitigate this problem by using a global hub with regional autonomous zones, where voting power for each zone are distributed based on a common geographic region. For instance, a common paradigm may be for individual cities, or regions, to operate their own zones while sharing a common hub (e.g. the Cosmos Hub), enabling municipal activity to persist in the event that the hub halts due to a temporary network partition. Note that this allows real geological, political, and network-topological features to be considered in designing robust federated fault-tolerant systems.

Federated Name Resolution System

NameCoin was one of the first blockchains to attempt to solve the name-resolution problem by adapting the Bitcoin blockchain. Unfortunately there have been several issues with this approach.

With Namecoin, we can verify that, for example, @satoshi was registered with a particular public key at some point in the past, but we wouldn't know whether the public key had since been updated recently unless we download all the blocks since the last update of that name. This is due to the limitation of Bitcoin's UTXO transaction Merkle-ization model, where only the transactions (but not mutable application state) are Merkle-ized into the block-hash. This lets us prove existence, but not the non-existence of later updates to a name. Thus, we can't know for certain the most recent value of a name without trusting a full node, or incurring significant costs in bandwidth by downloading the whole blockchain.

Even if a Merkle-ized search tree were implemented in NameCoin, its dependency on proof-of-work makes light client verification problematic. Light clients must download a complete copy of the headers for all blocks in the entire blockchain (or at least all the headers since the last update to a name). This means that the bandwidth requirements scale linearly with the amount of time [21]. In addition, name-changes on a proof-of-work blockchain requires waiting for additional proof-of-work confirmation blocks, which can take up to an hour on Bitcoin.

With Tendermint, all we need is the most recent block-hash signed by a quorum of validators (by voting power), and a Merkle proof to the current value associated with the name. This makes it possible to have a succinct, quick, and secure light-client verification of name values.

In Cosmos, we can take this concept and extend it further. Each name-registration zone in Cosmos can have an associated top-level-domain (TLD) name such as ".com" or ".org", and each name-

registration zone can have its own governance and registration rules.

Issuance and Incentives

The Atom Token

While the Cosmos Hub is a multi-asset distributed ledger, there is a special native token called the *atom*. Atoms are the only staking token of the Cosmos Hub. Atoms are a license for the holder to vote, validate, or delegate to other validators. Like Ethereum's ether, atoms can also be used to pay for transaction fees to mitigate spam. Additional inflationary atoms and block transaction fees are rewarded to validators and delegators who delegate to validators.

The `BurnAtomTx` transaction can be used to recover any proportionate amount of tokens from the reserve pool.

Fundraiser

The initial distribution of atom tokens and validators on Genesis will go to the donors of the Cosmos Fundraiser (75%), lead donors (5%), Cosmos Network Foundation (10%), and ALL IN BITS, Inc (10%). From genesis onward, 1/3 of the total amount of atoms will be rewarded to bonded validators and delegators every year.

See the [Cosmos Plan](#) for additional details.

Limitations on the Number of Validators

Unlike Bitcoin or other proof-of-work blockchains, a Tendermint blockchain gets slower with more validators due to the increased communication complexity. Fortunately, we can support enough validators to make for a robust globally distributed blockchain with very fast transaction confirmation times, and, as bandwidth,

storage, and parallel compute capacity increases, we will be able to support more validators in the future.

On genesis day, the maximum number of validators will be set to 100, and this number will increase at a rate of 13% for 10 years, and settle at 300 validators.

```
Year 0: 100
Year 1: 113
Year 2: 127
Year 3: 144
Year 4: 163
Year 5: 184
Year 6: 208
Year 7: 235
Year 8: 265
Year 9: 300
Year 10: 300
...

```

Becoming a Validator After Genesis Day

Atom holders who are not already can become validators by signing and submitting a `BondTx` transaction. The amount of atoms provided as collateral must be nonzero. Anyone can become a validator at any time, except when the size of the current validator set is greater than the maximum number of validators allowed. In that case, the transaction is only valid if the amount of atoms is greater than the amount of effective atoms held by the smallest validator, where effective atoms include delegated atoms. When a new validator replaces an existing validator in such a way, the existing validator becomes inactive and all the atoms and delegated atoms enter the unbonding state.

Penalties for Validators

There must be some penalty imposed on the validators for any intentional or unintentional deviation from the sanctioned protocol. Some evidence is immediately admissible, such as a double-sign at the same height and round, or a violation of

“prevote-the-lock” (a rule of the Tendermint consensus protocol). Such evidence will result in the validator losing its good standing and its bonded atoms as well its proportionate share of tokens in the reserve pool – collectively called its “stake” – will get slashed.

Sometimes, validators will not be available, either due to regional network disruptions, power failure, or other reasons. If, at any point in the past `ValidatorTimeoutWindow` blocks, a validator’s commit vote is not included in the blockchain more than `ValidatorTimeoutMaxAbsent` times, that validator will become inactive, and lose `ValidatorTimeoutPenalty` (DEFAULT 1%) of its stake.

Some “malicious” behavior does not produce obviously discernable evidence on the blockchain. In these cases, the validators can coordinate out of band to force the timeout of these malicious validators, if there is a supermajority consensus.

In situations where the Cosmos Hub halts due to a $\geq \frac{1}{3}$ coalition of voting power going offline, or in situations where a $\geq \frac{1}{3}$ coalition of voting power censor evidence of malicious behavior from entering the blockchain, the hub must recover with a hard-fork reorg-proposal. (Link to “Forks and Censorship Attacks”).

Transaction Fees

Cosmos Hub validators can accept any token type or combination of types as fees for processing a transaction. Each validator can subjectively set whatever exchange rate it wants, and choose whatever transactions it wants, as long as the `BlockGasLimit` is not exceeded. The collected fees, minus any taxes specified below, are redistributed to the bonded stakeholders in proportion to their bonded atoms, every `ValidatorPayoutPeriod` (DEFAULT 1 hour).

Of the collected transaction fees, **ReserveTax** (DEFAULT 2%) will go toward the reserve pool to increase the reserve pool and increase the security and value of the Cosmos network. These funds can also be distributed in accordance with the decisions made by the governance system.

Atom holders who delegate their voting power to other validators pay a commission to the delegated validator. The commission can be set by each validator.

Incentivizing Hackers

The security of the Cosmos Hub is a function of the security of the underlying validators and the choice of delegation by delegators. In order to encourage the discovery and early reporting of found vulnerabilities, the Cosmos Hub encourages hackers to publish successful exploits via a **ReportHackTx** transaction that says, “This validator got hacked. Please send bounty to this address”. Upon such an exploit, the validator and delegators will become inactive, **HackPunishmentRatio** (default 5%) of everyone’s atoms will get slashed, and **HackRewardRatio** (default 5%) of everyone’s atoms will get rewarded to the hacker’s bounty address. The validator must recover the remaining atoms by using their backup key.

In order to prevent this feature from being abused to transfer unvested atoms, the portion of vested vs unvested atoms of validators and delegators before and after the **ReportHackTx** will remain the same, and the hacker bounty will include some unvested atoms, if any.

Governance Specification

The Cosmos Hub is operated by a distributed organization that requires a well-defined governance mechanism in order to coordinate various changes to the blockchain, such as the variable

parameters of the system, as well as software upgrades and constitutional amendments.

All validators are responsible for voting on all proposals. Failing to vote on a proposal in a timely manner will result in the validator being deactivated automatically for a period of time called the **AbsenteeismPenaltyPeriod** (DEFAULT 1 week).

Delegators automatically inherit the vote of the delegated validator. This vote may be overridden manually. Unbonded atoms get no vote.

Each proposal requires a deposit of **MinimumProposalDeposit** tokens, which may be a combination of one or more tokens including atoms. For each proposal, the voters may vote to take the deposit. If more than half of the voters choose to take the deposit (e.g. because the proposal was spam), the deposit goes to the reserve pool, except any atoms which are burned.

For each proposal, voters may vote with the following options:

- Yea
- YeaWithForce
- Nay
- NayWithForce
- Abstain

A strict majority of Yea or YeaWithForce votes (or Nay or NayWithForce votes) is required for the proposal to be decided as passed (or decided as failed), but 1/3+ can veto the majority decision by voting “with force”. When a strict majority is vetoed, everyone gets punished by losing **VetoPenaltyFeeBlocks** (DEFAULT 1 day’s worth of blocks) worth of fees (except taxes which will not be affected), and the party that vetoed the majority

decision will be additionally punished by losing `VetoPenaltyAtoms` (DEFAULT 0.1%) of its atoms.

Parameter Change Proposal

Any of the parameters defined here can be changed with the passing of a `ParameterChangeProposal`.

Bounty Proposal

Atoms can be inflated and reserve pool funds spent with the passing of a `BountyProposal`.

Text Proposal

All other proposals, such as a proposal to upgrade the protocol, will be coordinated via the generic `TextProposal`.

Roadmap

See [the Plan](#).

Related Work

There have been many innovations in blockchain consensus and scalability in the past couple of years. This section provides a brief survey of a select number of important ones.

Consensus Systems

Classic Byzantine Fault Tolerance

Consensus in the presence of malicious participants is a problem dating back to the early 1980s, when Leslie Lamport coined the phrase “Byzantine fault” to refer to arbitrary process behavior that deviates from the intended behavior, in contrast to a “crash fault”, wherein a process simply crashes. Early solutions were discovered for synchronous networks where there is an upper bound on

message latency, though practical use was limited to highly controlled environments such as airplane controllers and datacenters synchronized via atomic clocks. It was not until the late 90s that Practical Byzantine Fault Tolerance (PBFT) [11] was introduced as an efficient partially synchronous consensus algorithm able to tolerate up to $\frac{1}{3}$ of processes behaving arbitrarily. PBFT became the standard algorithm, spawning many variations, including most recently one created by IBM as part of their contribution to Hyperledger.

The main benefit of Tendermint consensus over PBFT is that Tendermint has an improved and simplified underlying structure, some of which is a result of embracing the blockchain paradigm. Tendermint blocks must commit in order, which obviates the complexity and communication overhead associated with PBFT's view-changes. In Cosmos and many cryptocurrencies, there is no need to allow for block $N+i$ where $i \geq 1$ to commit, when block N itself hasn't yet committed. If bandwidth is the reason why block N hasn't committed in a Cosmos zone, then it doesn't help to use bandwidth sharing votes for blocks $N+i$. If a network partition or offline nodes is the reason why block N hasn't committed, then $N+i$ won't commit anyway.

In addition, the batching of transactions into blocks allows for regular Merkle-hashing of the application state, rather than periodic digests as with PBFT's checkpointing scheme. This allows for faster provable transaction commits for light-clients and faster inter-blockchain communication.

Tendermint Core also includes many optimizations and features that go above and beyond what is specified in PBFT. For example, the blocks proposed by validators are split into parts, Merkle-ized, and gossipped in such a way that improves broadcasting performance (see LibSwift [19] for inspiration). Also, Tendermint Core doesn't make any assumption about point-to-point

connectivity, and functions for as long as the P2P network is weakly connected.

BitShares delegated stake

While not the first to deploy proof-of-stake (PoS), BitShares1.0 [12] contributed considerably to research and adoption of PoS blockchains, particularly those known as “delegated” PoS. In BitShares, stake holders elect “witnesses”, responsible for ordering and committing transactions, and “delegates”, responsible for coordinating software updates and parameter changes.

BitShares2.0 aims to achieve high performance (100k tx/s, 1s latency) in ideal conditions, with each block signed by a single signer, and transaction finality taking quite a bit longer than the block interval. A canonical specification is still in development. Stakeholders can remove or replace misbehaving witnesses on a daily basis, but there is no significant collateral of witnesses or delegators in the likeness of Tendermint PoS that get slashed in the case of a successful double-spend attack.

Stellar

Building on an approach pioneered by Ripple, Stellar [13] refined a model of Federated Byzantine Agreement wherein the processes participating in consensus do not constitute a fixed and globally known set. Rather, each process node curates one or more “quorum slices”, each constituting a set of trusted processes. A “quorum” in Stellar is defined to be a set of nodes that contain at least one quorum slice for each node in the set, such that agreement can be reached.

The security of the Stellar mechanism relies on the assumption that the intersection of *any* two quorums is non-empty, while the availability of a node requires at least one of its quorum slices to consist entirely of correct nodes, creating a trade-off between using large or small quorum-slices that may be difficult to balance without imposing significant assumptions about trust. Ultimately,

nodes must somehow choose adequate quorum slices for there to be sufficient fault-tolerance (or any “intact nodes” at all, of which much of the results of the paper depend on), and the only provided strategy for ensuring such a configuration is hierarchical and similar to the Border Gateway Protocol (BGP), used by top-tier ISPs on the internet to establish global routing tables, and by that used by browsers to manage TLS certificates; both notorious for their insecurity.

The criticism in the Stellar paper of the Tendermint-based proof-of-stake systems is mitigated by the token strategy described here, wherein a new type of token called the *atom* is issued that represent claims to future portions of fees and rewards. The advantage of Tendermint-based proof-of-stake, then, is its relative simplicity, while still providing sufficient and provable security guarantees.

BitcoinNG

BitcoinNG is a proposed improvement to Bitcoin that would allow for forms of vertical scalability, such as increasing the block size, without the negative economic consequences typically associated with such a change, such as the disproportionately large impact on small miners. This improvement is achieved by separating leader election from transaction broadcast: leaders are first elected by proof-of-work in “micro-blocks”, and then able to broadcast transactions to be committed until a new micro-block is found. This reduces the bandwidth requirements necessary to win the PoW race, allowing small miners to more fairly compete, and allowing transactions to be committed more regularly by the last miner to find a micro-block.

Casper

Casper [16] is a proposed proof-of-stake consensus algorithm for Ethereum. Its prime mode of operation is “consensus-by-bet”. By letting validators iteratively bet on which block they believe will

become committed into the blockchain based on the other bets that they have seen so far, finality can be achieved eventually. [link](#). This is an active area of research by the Casper team. The challenge is in constructing a betting mechanism that can be proven to be an evolutionarily stable strategy. The main benefit of Casper as compared to Tendermint may be in offering “availability over consistency” – consensus does not require a $>2/3$ quorum of voting power – perhaps at the cost of commit speed or implementation complexity.

Horizontal Scaling

Interledger Protocol

The Interledger Protocol [14] is not strictly a scalability solution. It provides an ad hoc interoperation between different ledger systems through a loosely coupled bilateral relationship network. Like the Lightning Network, the purpose of ILP is to facilitate payments, but it specifically focuses on payments across disparate ledger types, and extends the atomic transaction mechanism to include not only hash-locks, but also a quorum of notaries (called the Atomic Transport Protocol). The latter mechanism for enforcing atomicity in inter-ledger transactions is similar to Tendermint’s light-client SPV mechanism, so an illustration of the distinction between ILP and Cosmos/IBC is warranted, and provided below.

1. The notaries of a connector in ILP do not support membership changes, and do not allow for flexible weighting between notaries. On the other hand, IBC is designed specifically for blockchains, where validators can have different weights, and where membership can change over the course of the blockchain.
2. As in the Lightning Network, the receiver of payment in ILP must be online to send a confirmation back to the sender. In a

token transfer over IBC, the validator-set of the receiver's blockchain is responsible for providing confirmation, not the receiving user.

3. The most striking difference is that ILP's connectors are not responsible or keeping authoritative state about payments, whereas in Cosmos, the validators of a hub are the authority of the state of IBC token transfers as well as the authority of the amount of tokens held by each zone (but not the amount of tokens held by each account within a zone). This is the fundamental innovation that allows for secure asymmetric transfer of tokens from zone to zone; the analog to ILP's connector in Cosmos is a persistent and maximally secure blockchain ledger, the Cosmos Hub.
4. The inter-ledger payments in ILP need to be backed by an exchange orderbook, as there is no asymmetric transfer of coins from one ledger to another, only the transfer of value or market equivalents.

Sidechains

Sidechains [15] are a proposed mechanism for scaling the Bitcoin network via alternative blockchains that are “two-way pegged” to the Bitcoin blockchain. (Two-way pegging is equivalent to bridging. In Cosmos we say "bridging" to distinguish from market-pegging). Sidechains allow bitcoins to effectively move from the Bitcoin blockchain to the sidechain and back, and allow for experimentation in new features on the sidechain. As in the Cosmos Hub, the sidechain and Bitcoin serve as light-clients of each other, using SPV proofs to determine when coins should be transferred to the sidechain and back. Of course, since Bitcoin uses proof-of-work, sidechains centered around Bitcoin suffer from the many problems and risks of proof-of-work as a consensus mechanism. Furthermore, this is a Bitcoin-maximalist solution that doesn't natively support a variety of tokens and

inter-zone network topology as Cosmos does. That said, the core mechanism of the two-way peg is in principle the same as that employed by the Cosmos network.

Ethereum Scalability Efforts

Ethereum is currently researching a number of different strategies to shard the state of the Ethereum blockchain to address scalability needs. These efforts have the goal of maintaining the abstraction layer offered by the current Ethereum Virtual Machine across the shared state space. Multiple research efforts are underway at this time. [\[18\]](#)[\[22\]](#)

Cosmos vs Ethereum 2.0 Mauve

Cosmos and Ethereum 2.0 Mauve [\[22\]](#) have different design goals.

- Cosmos is specifically about tokens. Mauve is about scaling general computation.
- Cosmos is not bound to the EVM, so even different VMs can interoperate.
- Cosmos lets the zone creator determine who validates the zone.
- Anyone can start a new zone in Cosmos (unless governance decides otherwise).
- The hub isolates zone failures so global token invariants are preserved.

General Scaling

Lightning Network

The Lightning Network is a proposed token transfer network operating at a layer above the Bitcoin blockchain (and other public blockchains), enabling improvement of many orders of magnitude in transaction throughput by moving the majority of transactions outside of the consensus ledger into so-called “payment channels”.

This is made possible by on-chain cryptocurrency scripts, which enable parties to enter into bilateral stateful contracts where the state can be updated by sharing digital signatures, and contracts can be closed by finally publishing evidence onto the blockchain, a mechanism first popularized by cross-chain atomic swaps. By opening payment channels with many parties, participants in the Lightning Network can become focal points for routing the payments of others, leading to a fully connected payment channel network, at the cost of capital being tied up on payment channels.

While the Lightning Network can also easily extend across multiple independent blockchains to allow for the transfer of value via an exchange market, it cannot be used to asymmetrically transfer tokens from one blockchain to another. The main benefit of the Cosmos network described here is to enable such direct token transfers. That said, we expect payment channels and the Lightning Network to become widely adopted along with our token transfer mechanism, for cost-saving and privacy reasons.

Segregated Witness

Segregated Witness is a Bitcoin improvement proposal [link](#) that aims to increase the per-block transaction throughput 2X or 3X, while simultaneously making block syncing faster for new nodes. The brilliance of this solution is in how it works within the limitations of Bitcoin's current protocol and allows for a soft-fork upgrade (i.e. clients with older versions of the software will continue to function after the upgrade). Tendermint, being a new protocol, has no design restrictions, so it has a different scaling priorities. Primarily, Tendermint uses a BFT round-robin algorithm based on cryptographic signatures instead of mining, which trivially allows horizontal scaling through multiple parallel blockchains, while regular, more frequent block commits allow for vertical scaling as well.

Appendix

Fork Accountability

A well designed consensus protocol should provide some guarantees in the event that the tolerance capacity is exceeded and the consensus fails. This is especially necessary in economic systems, where Byzantine behaviour can have substantial financial reward. The most important such guarantee is a form of *fork-accountability*, where the processes that caused the consensus to fail (ie. caused clients of the protocol to accept different values - a fork) can be identified and punished according to the rules of the protocol, or, possibly, the legal system. When the legal system is unreliable or excessively expensive to invoke, validators can be forced to make security deposits in order to participate, and those deposits can be revoked, or slashed, when malicious behaviour is detected [10].

Note this is unlike Bitcoin, where forking is a regular occurrence due to network asynchrony and the probabilistic nature of finding partial hash collisions. Since in many cases a malicious fork is indistinguishable from a fork due to asynchrony, Bitcoin cannot reliably implement fork-accountability, other than the implicit opportunity cost paid by miners for mining an orphaned block.

Tendermint Consensus

We call the voting stages *PreVote* and *PreCommit*. A vote can be for a particular block or for *Nil*. We call a collection of $>2/3$ *PreVotes* for a single block in the same round a *Polka*, and a collection of $>2/3$ *PreCommits* for a single block in the same round a *Commit*. If $>2/3$ *PreCommit* for *Nil* in the same round, they move to the next round.

Note that strict determinism in the protocol incurs a weak synchrony assumption as faulty leaders must be detected and

skipped. Thus, validators wait some amount of time, `TimeoutPropose`, before they `Prevote Nil`, and the value of `TimeoutPropose` increases with each round. Progression through the rest of a round is fully asynchronous, in that progress is only made once a validator hears from $>2/3$ of the network. In practice, it would take an extremely strong adversary to indefinitely thwart the weak synchrony assumption (causing the consensus to fail to ever commit a block), and doing so can be made even more difficult by using randomized values of `TimeoutPropose` on each validator.

An additional set of constraints, or Locking Rules, ensure that the network will eventually commit just one block at each height. Any malicious attempt to cause more than one block to be committed at a given height can be identified. First, a `PreCommit` for a block must come with justification, in the form of a Polka for that block. If the validator has already `PreCommit` a block at round R_1 , we say they are *locked* on that block, and the Polka used to justify the new `PreCommit` at round R_2 must come in a round R_polka where $R_1 < R_polka \leq R_2$. Second, validators must `Propose` and/or `PreVote` the block they are locked on. Together, these conditions ensure that a validator does not `PreCommit` without sufficient evidence as justification, and that validators which have already `PreCommit` cannot contribute to evidence to `PreCommit` something else. This ensures both safety and liveness of the consensus algorithm.

The full details of the protocol are described [here](#).

Tendermint Light Clients

The need to sync all block headers is eliminated in Tendermint-PoS as the existence of an alternative chain (a fork) means $\geq 1/3$ of bonded stake can be slashed. Of course, since slashing requires that someone share evidence of a fork, light clients should store any block-hash commits that it sees. Additionally, light clients

could periodically stay synced with changes to the validator set, in order to avoid [long range attacks](#) (but other solutions are possible).

In spirit similar to Ethereum, Tendermint enables applications to embed a global Merkle root hash in each block, allowing easily verifiable state queries for things like account balances, the value stored in a contract, or the existence of an unspent transaction output, depending on the nature of the application.

Preventing Long Range Attacks

Assuming a sufficiently resilient collection of broadcast networks and a static validator set, any fork in the blockchain can be detected and the deposits of the offending validators slashed. This innovation, first suggested by Vitalik Buterin in early 2014, solves the nothing-at-stake problem of other proof-of-stake cryptocurrencies (see [Related Work](#)). However, since validator sets must be able to change, over a long range of time the original validators may all become unbonded, and hence would be free to create a new chain from the genesis block, incurring no cost as they no longer have deposits locked up. This attack came to be known as the Long Range Attack (LRA), in contrast to a Short Range Attack, where validators who are currently bonded cause a fork and are hence punishable (assuming a fork-accountable BFT algorithm like Tendermint consensus). Long Range Attacks are often thought to be a critical blow to proof-of-stake.

Fortunately, the LRA can be mitigated as follows. First, for a validator to unbond (thereby recovering their collateral deposit and no longer earning fees to participate in the consensus), the deposit must be made untransferable for an amount of time known as the “unbonding period”, which may be on the order of weeks or months. Second, for a light client to be secure, the first time it connects to the network it must verify a recent block-hash against a trusted source, or preferably multiple sources. This

condition is sometimes referred to as “weak subjectivity”. Finally, to remain secure, it must sync up with the latest validator set at least as frequently as the length of the unbonding period. This ensures that the light client knows about changes to the validator set before a validator has its capital unbonded and thus no longer at stake, which would allow it to deceive the client by carrying out a long range attack by creating new blocks beginning back at a height where it was bonded (assuming it has control of sufficiently many of the early private keys).

Note that overcoming the LRA in this way requires an overhaul of the original security model of proof-of-work. In PoW, it is assumed that a light client can sync to the current height from the trusted genesis block at any time simply by processing the proof-of-work in every block header. To overcome the LRA, however, we require that a light client come online with some regularity to track changes in the validator set, and that the first time they come online they must be particularly careful to authenticate what they hear from the network against trusted sources. Of course, this latter requirement is similar to that of Bitcoin, where the protocol and software must also be obtained from a trusted source.

The above method for preventing LRA is well suited for validators and full nodes of a Tendermint-powered blockchain because these nodes are meant to remain connected to the network. The method is also suitable for light clients that can be expected to sync with the network frequently. However, for light clients that are not expected to have frequent access to the internet or the blockchain network, yet another solution can be used to overcome the LRA. Non-validator token holders can post their tokens as collateral with a very long unbonding period (e.g. much longer than the unbonding period for validators) and serve light clients with a secondary method of attesting to the validity of current and past block-hashes. While these tokens do not count toward the security of the blockchain’s consensus, they nevertheless can

provide strong guarantees for light clients. If historical block-hash querying were supported in Ethereum, anyone could bond their tokens in a specially designed smart contract and provide attestation services for pay, effectively creating a market for light-client LRA security.

Overcoming Forks and Censorship Attacks

Due to the definition of a block commit, any $\geq \frac{1}{3}$ coalition of voting power can halt the blockchain by going offline or not broadcasting their votes. Such a coalition can also censor particular transactions by rejecting blocks that include these transactions, though this would result in a significant proportion of block proposals to be rejected, which would slow down the rate of block commits of the blockchain, reducing its utility and value. The malicious coalition might also broadcast votes in a trickle so as to grind blockchain block commits to a near halt, or engage in any combination of these attacks. Finally, it can cause the blockchain to fork, by double-signing or violating the locking rules.

If a globally active adversary were also involved, it could partition the network in such a way that it may appear that the wrong subset of validators were responsible for the slowdown. This is not just a limitation of Tendermint, but rather a limitation of all consensus protocols whose network is potentially controlled by an active adversary.

For these types of attacks, a subset of the validators should coordinate through external means to sign a reorg-proposal that chooses a fork (and any evidence thereof) and the initial subset of validators with their signatures. Validators who sign such a reorg-proposal forego their collateral on all other forks. Clients should verify the signatures on the reorg-proposal, verify any evidence, and make a judgement or prompt the end-user for a decision. For example, a phone wallet app may prompt the user with a security

warning, while a refrigerator may accept any reorg-proposal signed by $\frac{1}{2}$ of the original validators by voting power.

No non-synchronous Byzantine fault-tolerant algorithm can come to consensus when $\geq \frac{1}{3}$ of voting power are dishonest, yet a fork assumes that $\geq \frac{1}{3}$ of voting power have already been dishonest by double-signing or lock-changing without justification. So, signing the reorg-proposal is a coordination problem that cannot be solved by any non-synchronous protocol (i.e. automatically, and without making assumptions about the reliability of the underlying network). For now, we leave the problem of reorg-proposal coordination to human coordination via social consensus on internet media. Validators must take care to ensure that there are no remaining network partitions prior to signing a reorg-proposal, to avoid situations where two conflicting reorg-proposals are signed.

Assuming that the external coordination medium and protocol is robust, it follows that forks are less of a concern than censorship attacks.

In addition to forks and censorship, which require $\geq \frac{1}{3}$ Byzantine voting power, a coalition of $> \frac{2}{3}$ voting power may commit arbitrary, invalid state. This is characteristic of any (BFT) consensus system. Unlike double-signing, which creates forks with easily verifiable evidence, detecting commitment of an invalid state requires non-validating peers to verify whole blocks, which implies that they keep a local copy of the state and execute each transaction, computing the state root independently for themselves. Once detected, the only way to handle such a failure is via social consensus. For instance, in situations where Bitcoin has failed, whether forking due to software bugs (as in March 2013), or committing invalid state due to Byzantine behavior of miners (as in July 2015), the well connected community of businesses, developers, miners, and other organizations established a social consensus as to what manual actions were

required by participants to heal the network. Furthermore, since validators of a Tendermint blockchain may be expected to be identifiable, commitment of an invalid state may even be punishable by law or some external jurisprudence, if desired.

ABCI Specification

ABCI consists of 3 primary message types that get delivered from the core to the application. The application replies with corresponding response messages.

The `AppendTx` message is the work horse of the application. Each transaction in the blockchain is delivered with this message. The application needs to validate each transactions received with the AppendTx message against the current state, application protocol, and the cryptographic credentials of the transaction. A validated transaction then needs to update the application state — by binding a value into a key values store, or by updating the UTXO database.

The `CheckTx` message is similar to AppendTx, but it's only for validating transactions. Tendermint Core's mempool first checks the validity of a transaction with CheckTx, and only relays valid transactions to its peers. Applications may check an incrementing nonce in the transaction and return an error upon CheckTx if the nonce is old.

The `Commit` message is used to compute a cryptographic commitment to the current application state, to be placed into the next block header. This has some handy properties. Inconsistencies in updating that state will now appear as blockchain forks which catches a whole class of programming errors. This also simplifies the development of secure lightweight clients, as Merkle-hash proofs can be verified by checking against the block-hash, and the block-hash is signed by a quorum of validators (by voting power).

Additional ABCI messages allow the application to keep track of and change the validator set, and for the application to receive the block information, such as the height and the commit votes.

ABCI requests/responses are simple Protobuf messages. Check out the [schema file](#).

AppendTx

- **Arguments:**

- `Data ([]byte)` : The request transaction bytes

- **Returns:**

- `Code (uint32)` : Response code
 - `Data ([]byte)` : Result bytes, if any
 - `Log (string)` : Debug or error message

- **Usage:**

Append and run a transaction. If the transaction is valid, returns CodeType.OK

CheckTx

- **Arguments:**

- `Data ([]byte)` : The request transaction bytes

- **Returns:**

- `Code (uint32)` : Response code
 - `Data ([]byte)` : Result bytes, if any
 - `Log (string)` : Debug or error message

- **Usage:**

Validate a transaction. This message should not mutate the state. Transactions are first run through CheckTx before broadcast to peers in the mempool layer. You can make CheckTx semi-stateful and clear the state upon `Commit` or `BeginBlock`, to allow for dependent sequences of transactions in the same block.

Commit

- **Returns:**
 - `Data ([]byte)` : The Merkle root hash
 - `Log (string)` : Debug or error message
- **Usage:**

Return a Merkle root hash of the application state.

Query

- **Arguments:**
 - `Data ([]byte)` : The query request bytes
- **Returns:**
 - `Code (uint32)` : Response code
 - `Data ([]byte)` : The query response bytes
 - `Log (string)` : Debug or error message

Flush

- **Usage:**

Flush the response queue. Applications that implement `types.Application` need not implement this message – it's handled by the project.

Info

- **Returns:**
 - `Data ([]byte)` : The info bytes
- **Usage:**

Return information about the application state. Application specific.

SetOption

- **Arguments:**
 - `Key (string)` : Key to set

- **Value (string)** : Value to set for key
- **Returns:**
 - **Log (string)** : Debug or error message
- **Usage:**

Set application options. E.g. Key=“mode”, Value=“mempool” for a mempool connection, or Key=“mode”, Value=“consensus” for a consensus connection. Other options are application specific.

InitChain

- **Arguments:**
 - **Validators ([]Validator)** : Initial genesis Validators
- **Usage:**

Called once upon genesis

BeginBlock

- **Arguments:**
 - **Height (uint64)** : The block height that is starting
- **Usage:**

Signals the beginning of a new block. Called prior to any AppendTxs.

EndBlock

- **Arguments:**
 - **Height (uint64)** : The block height that ended
- **Returns:**
 - **Validators ([]Validator)** : Changed validators with new voting powers (0 to remove)
- **Usage:**

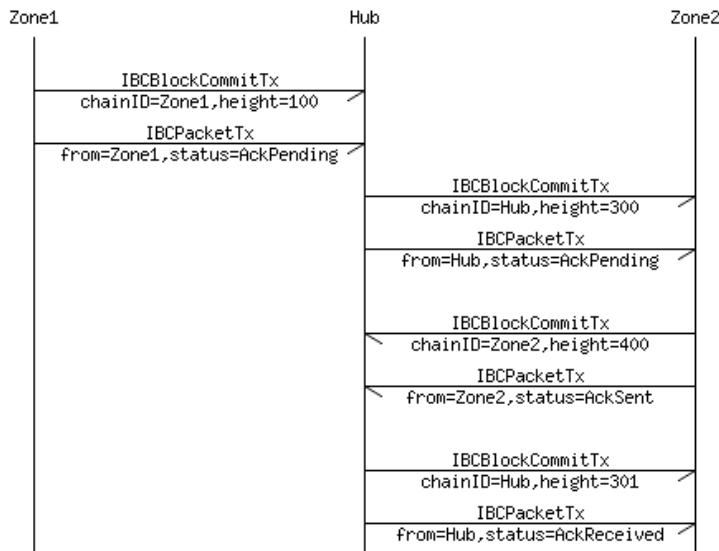
Signals the end of a block. Called prior to each Commit after all transactions

See [the ABCI repository](#) for more details.

IBC Packet Delivery Acknowledgement

There are several reasons why a sender may want the acknowledgement of delivery of a packet by the receiving chain. For example, the sender may not know the status of the destination chain, if it is expected to be faulty. Or, the sender may want to impose a timeout on the packet (with the `MaxHeight` packet field), while any destination chain may suffer from a denial-of-service attack with a sudden spike in the number of incoming packets.

In these cases, the sender can require delivery acknowledgement by setting the initial packet status to `AckPending`. Then, it is the receiving chain's responsibility to confirm delivery by including an abbreviated `IBCPacket` in the app Merkle hash.



First, an `IBCBLOCKCommit` and `IBCPACKETTx` are posted on “Hub” that proves the existence of an `IBCPacket` on “Zone1”. Say that `IBCPACKETTx` has the following value:

- `FromChainID` : “Zone1”
- `FromBlockHeight` : 100 (say)
- `Packet` : an `IBCPacket` :

- **Header** : an **IBCPacketHeader** :
 - **SrcChainID** : “Zone1”
 - **DstChainID** : “Zone2”
 - **Number** : 200 (say)
 - **Status** : **AckPending**
 - **Type** : “coin”
 - **MaxHeight** : 350 (say “Hub” is currently at height 300)
- **Payload** : <The bytes of a “coin” payload>

Next, an **IBCBlockCommit** and **IBCPacketTx** are posted on “Zone2” that proves the existence of an **IBCPacket** on “Hub”. Say that **IBCPacketTx** has the following value:

- **FromChainID** : “Hub”
- **FromBlockHeight** : 300
- **Packet** : an **IBCPacket** :
 - **Header** : an **IBCPacketHeader** :
 - **SrcChainID** : “Zone1”
 - **DstChainID** : “Zone2”
 - **Number** : 200
 - **Status** : **AckPending**
 - **Type** : “coin”
 - **MaxHeight** : 350
 - **Payload** : <The same bytes of a “coin” payload>

Next, “Zone2” must include in its app-hash an abbreviated packet that shows the new status of **AckSent**. An **IBCBlockCommit** and **IBCPacketTx** are posted back on “Hub” that proves the existence of an abbreviated **IBCPacket** on “Zone2”. Say that **IBCPacketTx** has the following value:

- **FromChainID** : “Zone2”

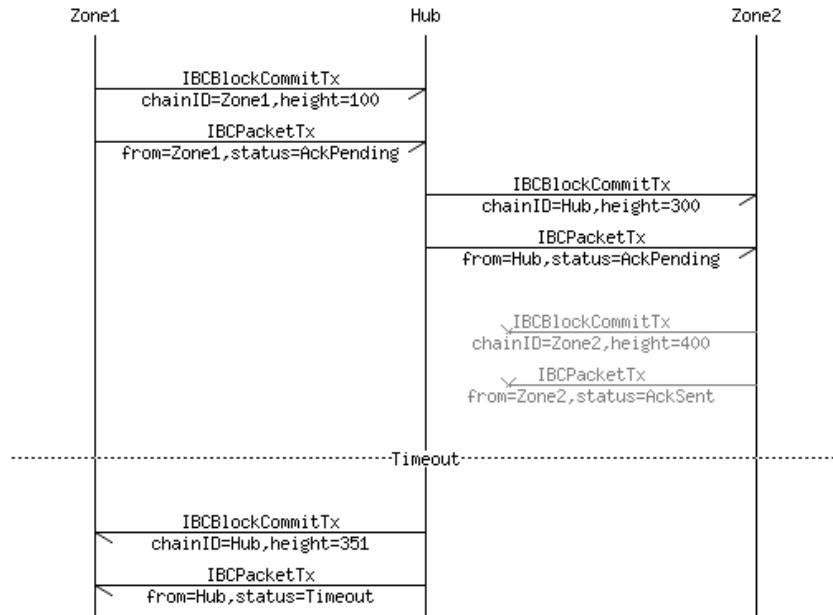
- **FromBlockHeight** : 400 (say)
- **Packet** : an **IBCPacket** :
 - **Header** : an **IBCPacketHeader** :
 - **SrcChainID** : "Zone1"
 - **DstChainID** : "Zone2"
 - **Number** : 200
 - **Status** : **AckSent**
 - **Type** : "coin"
 - **MaxHeight** : 350
 - **PayloadHash** : <The hash bytes of the same "coin" payload>

Finally, "Hub" must update the status of the packet from **AckPending** to **AckReceived**. Evidence of this new finalized status should go back to "Zone2". Say that **IBCPacketTx** has the following value:

- **FromChainID** : "Hub"
- **FromBlockHeight** : 301
- **Packet** : an **IBCPacket** :
 - **Header** : an **IBCPacketHeader** :
 - **SrcChainID** : "Zone1"
 - **DstChainID** : "Zone2"
 - **Number** : 200
 - **Status** : **AckReceived**
 - **Type** : "coin"
 - **MaxHeight** : 350
 - **PayloadHash** : <The hash bytes of the same "coin" payload>

Meanwhile, "Zone1" may optimistically assume successful delivery of a "coin" packet unless evidence to the contrary is proven on "Hub". In the example above, if "Hub" had not received an **AckSent**

status from “Zone2” by block 350, it would have set the status automatically to `Timeout`. This evidence of a timeout can get posted back on “Zone1”, and any tokens can be returned.

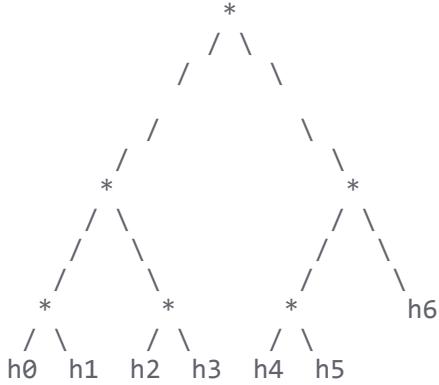


Merkle Tree & Proof Specification

There are two types of Merkle trees supported in the Tendermint/Cosmos ecosystem: The Simple Tree, and the IAVL+ Tree.

Simple Tree

The Simple Tree is a Merkle tree for a static list of elements. If the number of items is not a power of two, some leaves will be at different levels. Simple Tree tries to keep both sides of the tree the same height, but the left may be one greater. This Merkle tree is used to Merkle-ize the transactions of a block, and the top level elements of the application state root.



A SimpleTree with 7 elements

IAVL+ Tree

The purpose of the IAVL+ data structure is to provide persistent storage for key-value pairs in the application state such that a deterministic Merkle root hash can be computed efficiently. The tree is balanced using a variant of the [AVL algorithm](#), and all operations are $O(\log(n))$.

In an AVL tree, the heights of the two child subtrees of any node differ by at most one. Whenever this condition is violated upon an update, the tree is rebalanced by creating $O(\log(n))$ new nodes that point to unmodified nodes of the old tree. In the original AVL algorithm, inner nodes can also hold key-value pairs. The AVL+ algorithm (note the plus) modifies the AVL algorithm to keep all values on leaf nodes, while only using branch-nodes to store keys. This simplifies the algorithm while keeping the merkle hash trail short.

The AVL+ Tree is analogous to Ethereum's [Patricia tries](#). There are tradeoffs. Keys do not need to be hashed prior to insertion in IAVL+ trees, so this provides faster ordered iteration in the key space which may benefit some applications. The logic is simpler to implement, requiring only two types of nodes – inner nodes and leaf nodes. The Merkle proof is on average shorter, being a

balanced binary tree. On the other hand, the Merkle root of an IAVL+ tree depends on the order of updates.

We will support additional efficient Merkle trees, such as Ethereum's Patricia Trie when the binary variant becomes available.

Transaction Types

In the canonical implementation, transactions are streamed to the Cosmos hub application via the ABCI interface.

The Cosmos Hub will accept a number of primary transaction types, including `SendTx` , `BondTx` , `UnbondTx` , `ReportHackTx` , `SlashTx` , `BurnAtomTx` , `ProposalCreateTx` , and `ProposalVoteTx` , which are fairly self-explanatory and will be documented in a future revision of this paper. Here we document the two primary transaction types for IBC: `IBCBLOCKCommitTx` and `IBCPacketTx` .

IBCBLOCKCommitTx

An `IBCBLOCKCommitTx` transaction is composed of:

- **ChainID (string)** : The ID of the blockchain
- **BlockHash ([]byte)** : The block-hash bytes, the Merkle root which includes the app-hash
- **BlockPartsHeader (PartSetHeader)** : The block part-set header bytes, only needed to verify vote signatures
- **BlockHeight (int)** : The height of the commit
- **BlockRound (int)** : The round of the commit
- **Commit ([]Vote)** : The >2/3 Tendermint `Precommit` votes that comprise a block commit
- **ValidatorsHash ([]byte)** : A Merkle-tree root hash of the new validator set

- **ValidatorsHashProof (SimpleProof)** : A SimpleTree Merkle-proof for proving the **ValidatorsHash** against the **BlockHash**
- **AppHash ([]byte)** : A IAVLTree Merkle-tree root hash of the application state
- **AppHashProof (SimpleProof)** : A SimpleTree Merkle-proof for proving the **AppHash** against the **BlockHash**

IBCPacketTx

An **IBCPacket** is composed of:

- **Header (IBCPacketHeader)** : The packet header
- **Payload ([]byte)** : The bytes of the packet payload. *Optional*
- **PayloadHash ([]byte)** : The hash for the bytes of the packet. *Optional*

Either one of **Payload** or **PayloadHash** must be present. The hash of an **IBCPacket** is a simple Merkle root of the two items, **Header** and **Payload**. An **IBCPacket** without the full payload is called an *abbreviated packet*.

An **IBCPacketHeader** is composed of:

- **SrcChainID (string)** : The source blockchain ID
- **DstChainID (string)** : The destination blockchain ID
- **Number (int)** : A unique number for all packets
- **Status (enum)** : Can be one of **AckPending**, **AckSent**, **AckReceived**, **NoAck**, or **Timeout**
- **Type (string)** : The types are application-dependent. Cosmos reserves the "coin" packet type
- **MaxHeight (int)** : If status is not **NoAckWanted** or **AckReceived** by this height, status becomes **Timeout**. *Optional*

An **IBCPacketTx** transaction is composed of:

- **FromChainID (string)** : The ID of the blockchain which is providing this packet; not necessarily the source
- **FromBlockHeight (int)** : The blockchain height in which the following packet is included (Merkle-ized) in the block-hash of the source chain
- **Packet (IBCPacket)** : A packet of data, whose status may be one of **AckPending** , **AckSent** , **AckReceived** , **NoAck** , or **Timeout**
- **PacketProof (IAVLProof)** : A IAVLTree Merkle-proof for proving the packet's hash against the **AppHash** of the source chain at given height

The sequence for sending a packet from “Zone1” to “Zone2” through the “Hub” is depicted in {Figure X}. First, an **IBCPacketTx** proves to “Hub” that the packet is included in the app-state of “Zone1”. Then, another **IBCPacketTx** proves to “Zone2” that the packet is included in the app-state of “Hub”. During this procedure, the **IBCPacket** fields are identical: the **SrcChainID** is always “Zone1”, and the **DstChainID** is always “Zone2”.

The **PacketProof** must have the correct Merkle-proof path, as follows:

`IBC/<SrcChainID>/<DstChainID>/<Number>`

When “Zone1” wants to send a packet to “Zone2” through “Hub”, the **IBCPacket** data are identical whether the packet is Merkle-ized on “Zone1”, the “Hub”, or “Zone2”. The only mutable field is **Status** for tracking delivery.

Acknowledgements

We thank our friends and peers for assistance in conceptualizing, reviewing, and providing support for our work with Tendermint and Cosmos.

- [Zaki Manian](#) of [SkuChain](#) provided much help in formatting and wording, especially under the ABCI section
- [Jehan Tremback](#) of Althea and Dustin Byington for helping with initial iterations
- [Andrew Miller](#) of [Honey Badger](#) for feedback on consensus
- [Greg Slepak](#) for feedback on consensus and wording
- Also thanks to [Bill Gleim](#) and [Seunghwan Han](#) for various contributions.
- **Your name and organization here for your contribution**

Citations

- [1 Bitcoin:](#) <https://bitcoin.org/bitcoin.pdf>
- [2 ZeroCash:](#) <http://zerocash-project.org/paper>
- [3 Ethereum:](#) <https://github.com/ethereum/wiki/wiki/White-Paper>
- [4 TheDAO:](#)
<https://download.slock.it/public/DAO/WhitePaper.pdf>
- [5 Segregated Witness:](#)
<https://github.com/Bitcoin/bips/blob/master/bip-0141.mediawiki>
- [6 BitcoinNG:](#) <https://arxiv.org/pdf/1510.02037v2.pdf>
- [7 Lightning Network:](#) <https://lightning.network/lightning-network-paper-DRAFT-0.5.pdf>
- [8 Tendermint:](#)
<https://github.com/tendermint/tendermint/wiki>
- [9 FLP Impossibility:](#)
<https://groups.csail.mit.edu/tds/papers/Lynch/jacm85.pdf>
- [10 Slasher:](#) <https://blog.ethereum.org/2014/01/15/slasher-a-punitive-proof-of-stake-algorithm/>
- [11 PBFT:](#) <http://pmg.csail.mit.edu/papers/osdi99.pdf>
- [12 BitShares:](#) <https://bitshares.org/technology/delegated-proof-of-stake-consensus/>

- **13** Stellar: <https://www.stellar.org/papers/stellar-consensus-protocol.pdf>
- **14** Interledger: <https://interledger.org/rfcs/0001-interledger-architecture/>
- **15** Sidechains: <https://blockstream.com/sidechains.pdf>
- **16** Casper:
<https://blog.ethereum.org/2015/08/01/introducing-casper-friendly-ghost/>
- **17** ABCI: <https://github.com/tendermint/abci>
- **18** Ethereum Sharding:
<https://github.com/ethereum/EIPs/issues/53>
- **19** LibSwift:
<http://www.ds.ewi.tudelft.nl/fileadmin/pds/papers/PerformanceAnalysisOfLibswift.pdf>
- **20** DLS:
<http://groups.csail.mit.edu/tds/papers/Lynch/jacm88.pdf>
- **21** Thin Client Security:
https://en.bitcoin.it/wiki/Thin_Client_Security
- **22** Ethereum 2.0 Mauve Paper:
http://vitalik.ca/files/mauve_paper.html

Unsorted links

- <https://www.docdroid.net/ec7xGzs/31447721-ethereum-platform-review-opportunities-and-challenges-for-private-and-consortium-blockchains.pdf.html>
-

Cosmos Fundraiser
raised \$17 million in half an hour

 Fundraiser Ended

Join our [community chat](#).

Get Started

 Whitepaper

Get Newsletter



The Dai Stablecoin System

Whitepaper

<https://makerdao.com/>

By the Maker Team

December 2017

Overview of the Dai Stablecoin System	3
Collateralized Debt Position Smart Contracts	3
The CDP interaction process	4
Single-Collateral Dai vs Multi-Collateral Dai	4
Pooled Ether (Temporary mechanism for Single-Collateral Dai)	5
Price Stability Mechanisms	5
Target Price	5
Target Rate Feedback Mechanism	6
Sensitivity Parameter	7
Global Settlement	7
Global Settlement: Step by Step	7
Risk Management of The Maker Platform	8
Risk Parameters	9
MKR Token Governance	10
MKR and Multi-Collateral Dai	11
Automatic Liquidations of Risky CDPs	11
Liquidity Providing Contract (Temporary mechanism for Single-Collateral Dai)	12
Debt and Collateral Auctions (Multi-Collateral Dai)	12
Key External Actors	13
Keepers	13
Oracles	14
Global Settlers	14
Examples	14
Addressable Market	16
Risks and their Mitigation	17
Malicious hacking attack against the smart contract infrastructure	17
Black swan event in one or more collateral assets	18
Competition and the importance of ease-of-use	18
Pricing errors, irrationality and unforeseen events	18
Failure of centralized infrastructure	19
Conclusion	19
Glossary of Terms	20
Links	21

Overview of the Dai Stablecoin System

Popular digital assets such as Bitcoin (BTC) and Ether (ETH) are too volatile to be used as everyday currency. The value of a bitcoin often experiences large fluctuations, rising or falling by as much as 25% in a single day and occasionally rising over 300% in a month.¹.

The Dai Stablecoin is a collateral-backed cryptocurrency whose value is stable relative to the US Dollar. We believe that stable digital assets like Dai Stablecoin are essential to realizing the full potential of blockchain technology.

Maker is a smart contract platform on Ethereum that backs and stabilizes the value of Dai through a dynamic system of Collateralized Debt Positions (CDPs), autonomous feedback mechanisms, and appropriately incentivized external actors.

Maker enables anyone to leverage their Ethereum assets to generate Dai on the Maker Platform. Once generated, Dai can be used in the same manner as any other cryptocurrency: it can be freely sent to others, used as payments for goods and services, or held as long term savings. Importantly, the generation of Dai also creates the components needed for a robust decentralized margin trading platform.

Collateralized Debt Position Smart Contracts

Anyone who has collateral assets can leverage them to generate Dai on the Maker Platform through Maker's unique smart contracts known as Collateralized Debt Positions.²

CDPs hold collateral assets deposited by a user and permit this user to generate Dai, but generating also accrues debt. This debt effectively locks the deposited collateral assets inside the CDP until it is later covered by paying back an equivalent amount of Dai, at which point the owner can again withdraw their collateral . Active CDPs are always collateralized in excess, meaning that the value of the collateral is higher than the value of the debt.

¹ David Ernst [Hard Problems in Cryptocurrency](#)

² <https://github.com/makerdao>

The CDP interaction process

- **Step 1: Creating the CDP and depositing collateral**

The CDP user first sends a transaction to Maker to create the CDP, and then sends another transaction to fund it with the amount and type of collateral that will be used to generate Dai. At this point the CDP is considered collateralized.

- **Step 2: Generating Dai from the collateralized CDP**

The CDP user then sends a transaction to retrieve the amount of Dai they want from the CDP, and in return the CDP accrues an equivalent amount of debt, locking them out of access to the collateral until the outstanding debt is paid.

- **Step 3: Paying down the debt and Stability Fee**

When the user wants to retrieve their collateral, they have to pay down the debt in the CDP, plus the Stability fee that continuously accrue on the debt over time. The Stability Fee can only be paid in MKR. Once the user sends the requisite Dai and MKR to the CDP, paying down the debt and Stability Fee, the CDP becomes debt free.

- **Step 4: Withdrawing collateral and closing the CDP**

With the Debt and Stability Fee paid down, the CDP user can freely retrieve all or some of their collateral back to their wallet by sending a transaction to Maker.

Single-Collateral Dai vs Multi-Collateral Dai

Dai will initially launch with support for only one type of collateral, Pooled Ether. In the next 6-12 months we plan to upgrade Single-Collateral Dai to Multi-Collateral Dai. The primary difference is that it will support any number of CDP types.³

³ Mechanics that are temporarily in place in the system during the Single-Collateral phase are marked in this white paper

Pooled Ether (Temporary mechanism for Single-Collateral Dai)

At first, Pooled Ether (PETH) will be the only collateral type accepted on Maker. Users who wish to open a CDP and generate Dai during the first phase of the Maker Platform need to first obtain PETH. This is done instantly and easily on the blockchain by depositing ETH into a special smart contract that pools the ETH from all users, and gives them corresponding PETH in return.

If there is a sudden market crash in ETH, and a CDP ends up containing more debt than the value of its collateral, the Maker Platform automatically dilutes the PETH to recapitalize the system. This means that the proportional claim of each PETH goes down.

After the Maker Platform is upgraded to support multiple collateral types, PETH will be removed and replaced by ETH alongside the other new collateral types.

Price Stability Mechanisms

Target Price

The Dai Target Price has two primary functions on the Maker Platform: 1) It is used to calculate the collateral-to-debt ratio of a CDP, and 2) It is used to determine the value of collateral assets Dai holders receive in the case of a global settlement.

The Target Price is initially denominated in USD and starts at 1, translating to a 1:1 USD soft peg.

Target Rate Feedback Mechanism

In the event of severe market instability, the Target Rate Feedback Mechanism (TRFM) can be engaged. Engaging the TRFM breaks the fixed peg of Dai, but maintains the same denomination.

The TRFM is the automatic mechanism by which the Dai Stablecoin System adjusts the Target Rate in order to cause market forces to maintain stability of the Dai market price around the Target Price. The Target Rate determines the change of the Target Price over time, so it can act either as an incentive to hold Dai (if the Target Rate is positive) or an incentive to borrow Dai (If the Target Rate is negative). When the TRFM is not engaged the target rate is fixed at 0%, so the target price doesn't change over time and Dai is pegged.

When the TRFM is engaged, both the Target Rate and the Target Price change dynamically to balance the supply and demand of Dai by automatically adjusting user incentives for generating and holding Dai. The feedback mechanism pushes the market price of Dai towards the variable Target Price, dampening its volatility and providing real-time liquidity during demand shocks.

With the TRFM engaged, when the market price of Dai is below the Target Price, the Target Rate increases. This causes the Target Price to increase at a higher rate, causing generation of Dai with CDPs to become more expensive. At the same time, the increased Target Rate causes the capital gains from holding Dai to increase, leading to a corresponding increase in demand for Dai. This combination of reduced supply and increased demand causes the Dai market price to increase, pushing it back up towards the Target Price.

The same mechanism works in reverse if the Dai market price is higher than the Target Price: the Target Rate decreases, leading to an increased demand for generating Dai and a decreased demand for holding it. This causes the Dai market price to decrease, pushing it down towards the Target Price.

This mechanism is a negative feedback loop: Deviation away from the Target Price in one direction increases the force in the opposite direction.

Sensitivity Parameter

The TRFM's Sensitivity Parameter is a parameter that determines the magnitude of Target Rate change in response to Dai target/market price deviation. This tunes the rate of feedback to the scale of the system. MKR voters can set the Sensitivity Parameter but when

the TRFM is engaged the Target Price and the Target Rate are determined by market dynamics, and not directly controlled by MKR voters.

The Sensitivity Parameter is also what is used to engage or disengage the TRFM. If the Sensitivity Parameter and the Target Rate are both zero, Dai is pegged to the current Target Price.

Global Settlement

Global settlement is a process that can be used as a last resort to cryptographically guarantee the Target Price to holders of Dai. It shuts down and gracefully unwinds the Maker Platform while ensuring that all users, both Dai holders and CDP users, receive the net value of assets they are entitled to. The process is fully decentralized, and MKR voters govern access to it to ensure that it is only used in case of serious emergencies. Examples of serious emergencies are long term market irrationality, hacking or security breaches, and system upgrades.

Global Settlement: Step by Step

- **Step 1: Global Settlement is activated**

If enough actors who have been designated as global settlers by Maker Governance believe that the system is subject to a serious attack, or if a global settlement is scheduled as part of a technical upgrade, they can active the Global Settlement function. This stops CDP creation and manipulation, and freezes the Price Feed at a fixed value that is then used to process proportional claims for all users.

- **Step 2: Global Settlement claims are processed**

After Global Settlement has been activated, a period of time is needed to allow keepers to process the proportional claims of all Dai and CDP holders based on the fixed feed value. After this processing is done, all Dai holders and CDP holders will be able to claim a fixed amount of ETH with their Dai and CDPs.

- **Step 3: Dai and CDP holders claim the collateral with their Dai and CDPs**

Each Dai and CDP holder can call a claim function on the Maker Platform to exchange their Dai and CDPs directly for a fixed amount of ETH that corresponds to the calculated value of their assets, based on the target price of Dai.

E.g. If the Dai Target Price is 1 U.S. Dollar, The ETH/USD Price is 200 and a user holds 1000 Dai when Global Settlement is activated, after the processing period they will be able to claim exactly 5 ETH from the Maker Platform. There is no time limit for when the final claim can be made.

Risk Management of The Maker Platform

The MKR token allows holders to vote to perform the following Risk Management actions:

- **Add new CDP type:** Create a new CDP type with a unique set of Risk Parameters. A CDP type can either be a new type of collateral, or a new set of Risk Parameters for an existing collateral type.
- **Modify existing CDP types:** Change the Risk Parameters of one or more existing CDP types that were already added
- **Modify Sensitivity Parameter:** Change the sensitivity of the Target Rate Feedback Mechanism
- **Modify Target Rate:** Governance can change the Target Rate. In practice modifying the Target Rate will only be done in one specific circumstance: When MKR voters want to peg the price of Dai to its current Target Price. It will always be done in conjunction with modifying the Sensitivity Parameter. By setting both Sensitivity Parameter and Target Rate to 0%, the TRFM becomes disabled and the Target Price of Dai becomes pegged to its current value.

- **Choose the set of trusted oracles:** The Maker Platform derives its internal prices for collateral and the market price of Dai from a decentralized oracle infrastructure, consisting of a wide set of individual oracle nodes. MKR voters control how many nodes are in the set of trusted oracles, and who those nodes are. Up to half of the oracles can be compromised or malfunction without causing a disruption to the continued safe operation of the system
- **Modify Price Feed Sensitivity:** Change the rules that determine the largest change that the price feeds can affect on the internal price values in the system.
- **Choose the set of global settlers:** Global settlement is a crucial mechanic that allows the Maker Platform to survive attacks against the oracles or the governance process. The governance process chooses a set of global settlers and determines how many settlers are needed to activate global settlement.

Risk Parameters

Collateralized Debt Positions have multiple Risk Parameters that enforce how they can be used. Each CDP type has its own unique set of Risk Parameters, and these parameters are determined based on the risk profile of the collateral used by the CDP type. These parameters are directly controlled by MKR holders through voting, with one MKR giving its holder one vote.

The key Risk Parameters for CDPs are:

- **Debt Ceiling:** The Debt Ceiling is the maximum amount of debt that can be created by a single type of CDP. Once enough debt has been created by a CDP of any given type, it becomes impossible to create more unless existing CDPs are closed. The debt ceiling is used to ensure sufficient diversification of the collateral portfolio.
- **Liquidation Ratio:** The Liquidation Ratio is the collateral-to-debt ratio at which a CDP becomes vulnerable to Liquidation. A low Liquidation Ratio means MKR voters expect low price volatility of the collateral, while a high Liquidation Ratio means high volatility is expected.

- **Stability Fee:** The Stability Fee is a fee paid by every CDP. It is an annual percentage yield that is calculated on top of the existing debt of the CDP and has to be paid by the CDP user. The Stability Fee is denominated in Dai, but can only be paid using the MKR token. The amount of MKR that has to be paid is calculated based on a Price Feed of the MKR market price. When paid, the MKR is burned, permanently removing it from the supply.
- **Penalty Ratio:** The Penalty Ratio is used to determine the maximum amount of Dai raised from a Liquidation Auction that is used to buy up and remove MKR from the supply, with excess collateral getting returned to the CDP user who owned the CDP prior to its liquidation. The Penalty Ratio is used to cover the inefficiency of the liquidation mechanism. During the phase of Single-Collateral Dai, the Liquidation Penalty goes to buy and burn of PETH, benefitting the PETH to ETH ratio.

MKR Token Governance

In addition to payment of the Stability Fee on active CDPs, the MKR token plays an important role in the governance of the Maker Platform.

Governance is done at the system level through election of an Active Proposal by MKR voters. The Active Proposal is the smart contract that has been empowered by MKR voting to gain root access to modify the internal governance variables of the Maker Platform. Proposals can be in two forms: Single Action Proposal Contracts [SAPC], and Delegating Proposal Contracts [DPC].

Single Action Proposal Contracts are proposals that can only be executed once after gaining root access, and after execution immediately applies its changes to the internal governance variables of the Maker Platform. After the one-time execution, the SAPC deletes itself and cannot be re-used. This type of proposal is what will be used during the first phases of the system, as it is not very complicated to use, but is less flexible.

Delegating Proposal Contracts are proposals that continuously utilize their root access through second layer governance logic that is codified inside the DPC. The second layer governance logic can be relatively simple, such as defining a protocol for holding a weekly

vote on updated risk parameters. It can also implement more advanced logic, such as restrictions on the magnitude of governance actions within defined time periods, or even delegating some or all of its permissions further to one or more third layer DPCs with or without restrictions.

Any Ethereum account can deploy valid proposal smart contracts. MKR voters can then use their MKR tokens to cast approval votes for one or more proposals that they want to elect as the Active Proposal. The smart contract that has the highest total number of approval votes from MKR voters is elected as the Active Proposal.

MKR and Multi-Collateral Dai

After the upgrade to Multi-Collateral Dai, MKR will take on a more significant role in the Dai Stablecoin System by replacing PETH as the the recapitalization resource. When CDPs become undercollateralized due to market crashes, the MKR supply is automatically diluted and sold off in order to raise enough funds to recapitalize the system.

Automatic Liquidations of risky CDPs

To ensure there is always enough collateral in the system to cover the value of all outstanding Debt (according to the Target Price), a CDP can be liquidated if it is deemed to be too risky. The Maker Platform determines when to liquidate a CDP by comparing the Liquidation Ratio with the current collateral-to-debt ratio of the CDP.

Each CDP type has its own unique Liquidation Ratio that is controlled by MKR voters and established based on the risk profile of the particular collateral asset of that CDP type.

Liquidation occurs when a CDP hits its Liquidation Ratio. The Maker Platform will automatically buy the collateral of the CDP and subsequently sell it off. There is a temporary mechanism in place for Single-Collateral Dai called a Liquidity Providing Contract. For Multi-Collateral Dai an auction mechanism will be used.

Liquidity Providing Contract (Temporary mechanism for Single-Collateral Dai)

During Single-Collateral Dai, the mechanism for liquidation is a Liquidity Providing Contract: a smart contract that trades directly with ethereum users and keepers according to the price feed of the system.

When a CDP is liquidated, it is immediately acquired by the system. The CDP owner receives the value of the leftover collateral minus the debt, Stability Fee and Liquidation Penalty.

The PETH collateral is set for sale in the Liquidity Providing Contract, and keepers can atomically purchase the PETH by paying Dai. All Dai paid this way are immediately removed from the Dai supply, until an amount equal to the CDP debt has been removed. If any Dai is paid in excess of the debt shortfall, the excess Dai is used to purchase PETH from the market and burn it, which positively changes the ETH to PETH ratio. This results in a net value gain for PETH holders.

If the PETH sell-off initially does not raise enough Dai to cover the entire debt shortfall, more PETH is continuously created and sold off. New PETH created this way negatively changes the ETH to PETH ratio, causing PETH holders to lose value.

Debt and Collateral Auctions (Multi-Collateral Dai)

During a liquidation, the Maker platform buys the collateral of a CDP and subsequently sells it in an automatic auction. This auction mechanism enables the system to settle CDPs even when price information is unavailable.

In order to take over the collateral of the CDP so that it can be sold, the system first needs to raise enough Dai to cover the CDP's debt. This is called a Debt Auction, and works by diluting the supply of the MKR token and selling it to bidders in an auction format.

In parallel, the collateral of the CDP is sold in a Collateral Auction where all proceeds (also denominated in Dai) up to the CDP debt amount plus a Liquidation Penalty (A Risk Parameter determined by MKR voting) is used to buy MKR and remove it from the supply. This directly counteracts the MKR dilution that happened during the Debt Auction. If enough Dai is bid to fully cover the CDP debt plus the Liquidation Penalty, the Collateral Auction switches to a reverse auction mechanism and tries to sell as little collateral as possible--any leftover collateral is returned to the original owner of the CDP.

Key External Actors

In addition to its smart contract infrastructure, the Maker Platform relies on certain external actors to maintain operations. Keepers are external actors who take advantage of the economic incentives presented by the Maker platform. Oracles and Global Settlers are external actors with special permissions in the system assigned to them by MKR voters.

Keepers

A keeper is an independent (usually automated) actor that is incentivized by profit opportunities to contribute to decentralized systems. In the context of the Dai Stablecoin System, keepers participate in the Debt Auctions and Collateral Auctions when CDPs are liquidated.

Keepers also trade Dai around the Target Price. Keepers sell Dai when the market price is higher than the Target Price and buy Dai when the market price is below the Target Price to profit from the expected long-term convergence towards the Target Price.

Oracles

The Maker Platform requires real time information about the market price of the assets used as collateral in CDPs in order to know when to trigger liquidations. The Maker Platform also needs information about the market price of Dai and its deviation from the Target Price in order to adjust the Target Rate when the TRFM is engaged. MKR voters choose a set of trusted oracles to feed this information to the Maker Platform through Ethereum transactions.

To protect the system from an attacker who gains control of a majority of the oracles, and from other forms of collusion, there is a global variable that determines the maximum change to the value of the price feed permitted by the system. This variable is known as the Price Feed Sensitivity Parameter.

As an example of how the Price Feed Sensitivity Parameter works, if the Price Feed Sensitivity Parameter is defined as “5% in 15 minutes”, the price feeds cannot change more than 5% within one 15 minute period, and changing ~15% would take 45 minutes. This restriction ensures there is enough time to trigger a global settlement in the event that an attacker gains control over a majority of the oracles.

Global Settlers

Global Settlers are external actors similar to price feed oracles and are the last line of defense for the Dai Stablecoin System in the event of an attack. The set of global settlers, selected by MKR voters, have the authority to trigger global settlement. Aside from this authority, these actors do not have any additional special access or control within the system.

Examples

The Dai Stablecoin System can be used by anyone without any restrictions or sign-up process.

- **Example 1:** Bob needs a loan, so he decides to generate 100 Dai. He locks an amount of ETH worth significantly more than 100 Dai into a CDP and uses it to generate 100 Dai. The 100 Dai is instantly sent directly to his Ethereum account. Assuming that the Stability Fee is 1% per year, Bob will need 101 Dai to cover the CDP if he decides to retrieve his ETH one year later.

One of the primary use cases of CDPs is margin trading by CDP users.

- **Example 2:** Bob wishes to go margin long on the ETH/Dai pair, so he generates 100 USD worth of Dai by posting 150 USD worth of ETH to a CDP. He then buys another 100 USD worth of ETH with his newly generated Dai, giving him a net 1.66x ETH/USD exposure. He's free to do whatever he wants with the 100 USD worth of ETH he obtained by selling the Dai. The original ETH collateral (150 USD worth) remains locked in the CDP until the debt plus the Stability Fee is covered.

Although CDPs are not fungible with each other, the ownership of a CDP is transferable. This allows CDPs to be used in smart contracts that perform more complex methods of Dai generation (for example, involving more than one actor).

- **Example 3:** Alice and Bob collaborate using an Ethereum OTC contract to issue 100 USD worth of Dai backed by ETH. Alice contributes 50 USD worth of ETH, while Bob contributes 100 USD worth. The OTC contract takes the funds and creates a CDP, thus generating 100 USD worth of Dai. The newly generated Dai are automatically sent to Bob. From Bob's point of view, he is buying 100 USD worth of Dai by paying the equivalent value in ETH. The contract then transfers ownership of the CDP to Alice. She ends up with 100 USD worth of debt (denominated in Dai) and 150 USD worth of collateral (denominated in ETH). Since she started with only 50 USD worth of ETH, she is now 3x leveraged long ETH/USD.

Liquidations ensure that in the event of a price crash of the collateral backing a CDP type, the system will automatically be able to close CDPs that become too risky. This ensures that the outstanding Dai supply remains fully collateralized.

- **Example 4:** Let's assume that there is an Ether CDP type with a Liquidation Ratio of 145%, a Penalty Ratio of 105%, and we have an Ether CDP with a collateral-to-debt ratio of 150%. The Ether price now crashes 10% against the Target Price, causing the collateral-to-debt ratio of the CDP to fall to ~135%. As it falls below the Liquidation Ratio, traders can trigger its Liquidation and begin bidding with Dai for buying MKR in the debt auction. Simultaneously, traders can begin bidding with Dai for buying the ~135 Dai worth of collateral in the collateral auction. Once there is at least 105 Dai being bid on the Ether collateral, traders reverse bid to take the least amount of collateral for 105 Dai. Any remaining collateral is returned to the CDP owner.

Addressable Market

As mentioned in the introduction, a cryptocurrency with price stability is a basic requirement for the majority of decentralized applications. As such, the potential market for Dai is at least as large as that of the entire blockchain industry. The following is a short, non-exhaustive list of some of the immediate markets (in both the blockchain and the wider industry) for the Dai Stablecoin System in its capacity as a cryptocurrency with price stability and its use case as a decentralized margin trading platform:

- **Prediction Markets & Gambling Applications:** When making an unrelated prediction, it is obvious not to want to increase one's risk by placing the bet using a volatile cryptocurrency. Long term bets become especially infeasible if the user has to also gamble on the future price of the volatile asset used to place the bet. Instead, a cryptocurrency with price stability like Dai will be the natural choice for prediction market and gambling users.
- **Financial Markets; Hedging, Derivatives, Leverage:** CDPs will allow for permissionless leveraged trading. Dai will also be useful as stable and reliable collateral in custom derivative smart contracts, such as options or CFD's.
- **Merchant receipts, Cross-border transactions and remittances:** Foreign exchange volatility mitigation and a lack of intermediaries means the transaction costs of international trade can be significantly reduced by using Dai.
- **Transparent accounting systems:** Charities, NGO's and Governments will all see increases in efficiency and lower levels of corruption by utilizing Dai.

Risks and their Mitigation

There are many potential risks facing the successful development, deployment, and operation of the Maker Platform. It is vital that the Maker community takes all necessary steps to mitigate these risks. The following is a list spells out some of the risks identified and the accompanying plan for risk mitigation:

Malicious hacking attack against the smart contract infrastructure

The greatest risk to the system during its early stages is the risk of a malicious programmer finding an exploit in the deployed smart contracts, and using it to break or steal from the system before the vulnerability can be fixed. In a worst case scenario, all decentralized digital assets that are held as collateral in The Maker Platform, such as Ether (ETH) or Augur Reputation (REP), could be stolen without any chance of recovery. *The part of the collateral portfolio that is not decentralized, such as Digix Gold IOU's, would not be stolen in such an event as they can be frozen and controlled through a centralized backdoor.*

Mitigation: Smart contract security and best security practices have been the absolute highest priority of the Dai development effort since its inception. The codebase has already undergone three independent security audits by some of the best security researchers in the blockchain industry.

In the very long term, the risk of getting hacked can theoretically be almost completely mitigated through formal verification of the code. This means mathematically proving that the code does exactly what it is intended to do. While complete formal verification is a very long term goal, significant work towards it has already been completed, including a full reference implementation of the Dai Stablecoin System in the functional programming language Haskell, which serves as a stepping stone towards more sophisticated formalizations that are currently under active research and development

Black swan event in one or more collateral assets

Another high impact risk is a potential Black Swan event on collateral used for the Dai. This could either happen in the early stages of Dai Stablecoin System, before MKR is robust enough to support inflationary dilutions, or after the Dai Stablecoin System supports a diverse portfolio of collateral.

Mitigation: CDP collateral will be limited to ETH in the early stages, with the debt ceiling initially limited and growing gradually over time.

Competition and the importance of ease-of-use

As mentioned previously, there is a large amount of money and brainpower working on cryptocurrency with price stability. By virtue of having “true decentralization”, the Dai Stablecoin System is by far the most complex model being contemplated in the blockchain industry. A perceived risk is a movement among cryptocurrency users where the ideals of decentralization are exchanged for the simplicity and marketing of centralized digital assets.

Mitigation: We expect that Dai will be very easy to use for a regular cryptocurrency user. Dai will be a standard Ethereum token adhering to the ERC-20 standard and will be readily available with high liquidity across the ecosystem. Dai has been designed in such a way that the average user need not understand the underlying mechanics of the system in order to use it.

The complexities of the Dai Stablecoin System will need to be understood primarily by Keepers and capital investment companies that use the Dai Stablecoin System for margin trading. These types of users have enough resources to onboard themselves as long as there is abundant and clear documentation of every aspect of the system's mechanics. The Maker community will ensure that this is the case.

Pricing errors, irrationality and unforeseen events

A number of unforeseen events could potentially occur, such as a problem with the price feed from the Oracles, or irrational market dynamics that cause variation in the value of Dai for an extended period of time. If confidence is lost in the system, the TRFM adjustments or even MKR dilution could reach extreme levels while still not bringing enough liquidity and stability to the market.

Mitigation: The Maker community will need to incentivize a sufficiently large capital pool to act as Keepers of the market in order to maximize rationality and market efficiency and allow the Dai supply to grow at a steady pace without major market shocks.

Failure of centralized infrastructure

The Maker Team plays a major role in the development and governance of the Maker Platform in its early days: budgeting for expenses, hiring new developers, seeking partnerships and institutional users, and interfacing with regulators and other key external stakeholders. Should the Maker Team fail in some capacity — for legal reasons, or due to internal problems with management — the future of Maker could be at risk without a proper backup plan.

Mitigation: The Maker community exists partly to act as the decentralized counterparty to the Maker Team. It is a loose collective of independent actors who are all aligned by holding the MKR token, giving them a strong incentive to see the Maker Platform succeed. During the early phases of MKR distribution, great care was taken to ensure that the most important core developers received a significant MKR stake. In the event that the Maker Team is no longer effectively able to lead the development of the Maker Platform, individual MKR holders will be incentivized to fund developers (or simply carry out development themselves) in an effort to protect their investment.

Conclusion

The Dai Stablecoin System was designed to solve the crucial problem of stable exchange of value in the Ethereum ecosystem and the wider blockchain economy. We believe that the mechanism through which Dai is created, transacted, and retired, along with the direct Risk Management role of MKR holders, will allow for self-interested Keepers to maintain the price stability of Dai over time in an efficient manner. The founders of the Maker community have established a prudent governance roadmap that is appropriate for the needs of agile development in the short term, but also coherent with the ideals of decentralization over time. The development roadmap is aggressive and focused on widespread adoption of Dai in a responsible fashion.

Glossary of Terms

- **Collateralized Debt Position (CDP):** A smart contract whose users receive an asset (Dai), which effectively operates as a debt instrument with an interest rate. The CDP user has posted collateral in excess of the value of the loan in order to guarantee their debt position.
- **Dai:** The cryptocurrency with price stability that is the asset of exchange in the Dai Stablecoin System. It is a standard Ethereum token adhering to the ERC20 standard.
- **Debt Auction:** The reverse auction selling MKR for Dai to cover Emergency Debt when a CDP becomes undercollateralized.
- **Collateral Auction:** The auction selling collateral from a CDP undergoing liquidation. It is designed to prioritize covering the debt owed by the CDP, and secondarily to give the CDP owner the best possible price for their excess collateral refund.
- **The Dai Foundation:** A decentralized team of smart contract developers committed to the development and successful launch of the Maker Platform.
- **Keepers:** Independent economic actors that trade Dai, CDPs and/or MKR; create Dai or close CDPs; and seek arbitrage on The Dai Stablecoin System. As a result, Keepers help maintain Dai market rationality and price stability.
- **MKR:** The ERC20 token used by MKR voters for voting. It also serves as a backstop in the case of insolvent CDPs.
- **MKR Voters:** MKR holders who actively manage the risk of the Dai Stablecoin System by voting on Risk Parameters.
- **Maker:** The name of the Decentralized Autonomous Organization that is made up of the Maker Platform technical infrastructure, and the community of MKR voters.

- **Oracles:** Ethereum accounts (either contracts or users) selected to provide price feeds into various components of Maker Platform.
- **Risk Parameters:** The variables that determine (among other things) when the Maker Platform automatically judges a CDP to be Risky, allowing Keepers to liquidate it.
- **Sensitivity Parameter:** The variable that determines how aggressively the Dai Stablecoin System automatically changes the Target Rate in response to Dai market price deviations.
- **Target Rate Feedback Mechanism (TRFM):** The automatic mechanism by which the Dai Stablecoin System adjusts the Target Rate in order to cause market forces to maintain stability of the Dai market price around the Target Price.

Links

- **Chat:** <https://chat.makerdao.com/> — Primary platform of community interaction
- **Forum:** <https://forum.makerdao.com/> — For debate and proposals
- **Subreddit:** <https://reddit.com/r/makerdao/> — Best place to get latest news and links
- **GitHub:** <https://github.com/makerdao/> — Repository of the public Maker code
- **TeamSpeak:** <https://ts.makerdao.com/> — For governance meeting conference calls
- **SoundCloud:** <https://soundcloud.com/makerdao/> — Governance meeting recordings
- **Oasis:** <https://oasisdex.com/> — MKR and Dai decentralized exchange
- **Sai:** <https://sai.makerdao.com/> — Experimental stablecoin

W
h
it
e
p
a
p
e
r

Nathan Marley edited this page on 23 Aug 2018 · [11 revisions](#)

Dash: A Payments-Focused Cryptocurrency

Evan Duffield - evan@dash.org

Daniel Diaz - daniel@dash.org

Abstract. *A cryptocurrency based on Bitcoin, the work of Satoshi Nakamoto, with various improvements such as a two-tier incentivized network, known as the masternode network. Included are other improvements such as PrivateSend, for increasing fungibility, and InstantSend, which allows instant transaction confirmation without a centralized authority.*

1 Introduction

Bitcoin [1] is a cryptocurrency that has emerged as a popular medium of exchange and is the first digital currency that has attracted a substantial number of users [2]. Since its inception in 2009, Bitcoin has been rapidly growing in mainstream adoption and merchant usage [3]. A main issue with the acceptance of Bitcoin in point-of-sale (POS) situations is the time required to wait for the network to confirm the transaction made is valid. Some payment processors have created methods to allow vendors to take zero-confirmation transactions, but these solutions utilize a trusted counterparty to mediate the transaction outside of the protocol.

Bitcoin provides pseudonymous transactions in a public ledger, with a one-to-one relationship between sender and receiver. This provides a permanent record of all transactions that have ever taken place on the network [5]. Bitcoin is widely known in academic circles to provide a low level of privacy, although with this limitation many people still entrust their financial history to its blockchain.

Dash is the first cryptocurrency based on the work of Satoshi Nakamoto with built-in privacy functions. In this paper we propose a series of improvements to Bitcoin resulting in a decentralized, strongly anonymous cryptocurrency, with tamper-proof instant transactions and a secondary peer-to-peer (P2P) network incentivized to provide services to the Dash Network.

2 Masternode Network

Full nodes are servers running on a P2P network that allow peers to use them to receive updates about the events on the network. These nodes utilize significant amounts of traffic and other resources that incur a substantial cost. As a result, a steady decrease in the amount of these nodes has been observed for some time on the Bitcoin network [7] and as a result, block propagation times have been upwards of 40 seconds [14]. Many solutions have been proposed such as a new reward scheme by Microsoft Research [4] and the Bitnodes incentive program [6].

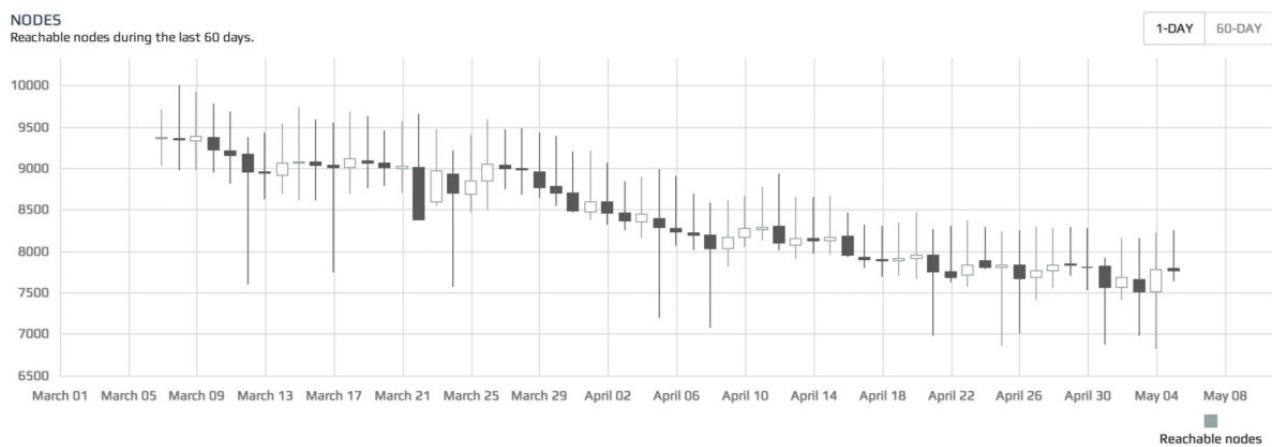


Figure 1: Bitcoin Full nodes in the spring of 2014

These nodes are very important to the health of the network. They provide clients with the ability to synchronize and facilitate quick propagation of messages throughout the network. We propose adding a secondary network, known as the Dash masternode network. These nodes will have high availability and provide a required level of service to the network in order to take part in the Masternode Reward Program.

2.1 Masternode Reward Program - Cost and Payments

Much of the reason for the decrease of full nodes on the Bitcoin network is the lack of incentive to run one. Over time, the cost of running a full node increases as the network gets used more, creating more bandwidth and costing the operator more money. As the cost rises, operators consolidate their services to be cheaper to run, or run a light client which does not help the network at all.

Masternodes are full nodes, just like in the Bitcoin network, except they must provide a level of service to the network and have a bond of collateral to participate. The collateral is never forfeit and is safe while the masternode is operating. This permits masternode operators to provide a service to the network, earn payment for their services and reduce the volatility of the currency.

To run a masternode, the operator must demonstrate control over 1,000 DASH. When active, masternodes provide services to clients on the network, and in return receive regular payment from the block reward. Like miners, masternodes are all paid from the block reward, 45% of which is dedicated to this program.

Due to the fact that the masternode rewards program is a fixed percentage and the masternode network nodes are fluctuating, expected masternode rewards will vary according to the current total count of active masternodes. Payments for a standard day for running a masternode can be calculated by using the following formula:

$$(n / t) * r * b * a$$

Where:

n is the number of masternodes an operator controls

t is the total number of masternodes

r is the current block reward (presently averaging about 3.3 DASH)
 b is blocks in an average day. For the Dash network this usually is 576.

a is the average masternode payment (45% of the average block reward)

The cost associated with running a masternode creates a hard and soft limit of active nodes on the network. Currently with 8.2 million DASH in circulation, only 8,200 nodes could possibly be running on the network. The soft limit is imposed by the price it costs to acquire a node and the limited liquidity on exchanges due to usage of Dash as a currency and not merely an investment.

2.2 Deterministic Ordering

A special deterministic algorithm is used to create a pseudo-random ordering of the masternodes. By using the hash from the proof-of-work for each block, security of this functionality will be provided by the mining network.

Pseudocode, for selecting a masternode:

```
For(mastenode in masternodes){  
    current_score = mastenode.CalculateScore();  
  
    if(current_score > best_score){  
        best_score = current_score;  
        winning_node = mastenode;  
    }  
}  
  
CMasterNode::CalculateScore(){  
    pow_hash = GetProofOfWorkHash(nBlockHeight); // get the  
hash of this block
```

```
    pow_hash_hash = Hash(pow_hash); //hash the P0W hash to  
increase the entropy  
    difference = abs(pow_hash_hash - masternode_vin);  
    return difference;  
}
```

The example code can be extended further to provide rankings of masternodes also, a “second”, “third”, “fourth” masternode in the list to be selected.

2.3 Trustless Quorums

Currently the Dash network has ~4,800 active masternodes [8]. By requiring 1,000DASH collateral to become an active masternode, we create a system in which no one can control the entire network of masternodes. For example, if someone wanted to control 50% of the masternode network, they would have to buy 4,800,000 DASH from the open market. This would raise the price substantially and it would become impossible to acquire the needed DASH.

With the addition of the masternode network and the collateral requirements, we can use this secondary network to do highly sensitive tasks in a trustless way, where no single entity can control the outcome. By selecting N pseudo random masternodes from the total pool to perform the same task, these nodes can act as an oracle, without having the whole network do the task.

As an example, implementation of a trustless quorum (see InstantSend [9]), which uses quorums to approve transactions and lock the inputs or the proof-of-service implementation [10].

Another example use for trustless quorums can include utilizing the masternode network as a decentralized oracle for financial markets,

making secure decentralized contracts a possibility. As an example contract, if Apple Stock (AAPL) is over \$300 on Dec 31, 2018 pay public key A, otherwise pay public key B.

2.4 Roles and Proof-Of-Service

Masternodes can provide any number of extra services to the network. As a proof-of-concept, our first implementation included PrivateSend and InstantSend. By utilizing what we call proof-of-service, we can require that these nodes are online, responding and even at the correct block height.

Bad actors could also run masternodes, but not provide any of the quality service that is required of the rest of the network. To reduce the possibility of people using the system to their advantage nodes must ping the rest of the network to ensure they remain active. This work is done by the masternode network by selecting 2 quorums per block. Quorum A checks the service of Quorum B each block. Quorum A are the closest nodes to the current block hash, while Quorum B are the furthest nodes from said hash.

Masternode A (1) checks Masternode B (rank 2300)

Masternode A (2) checks Masternode B (rank 2299)

Masternode A (3) checks Masternode B (rank 2298)

All work done to check the network to prove that nodes are active is done by the masternode network itself. Approximately 1% of the network will be checked each block. This results in the entire network being checked about six times per day. In order to keep this system trustless, we select nodes randomly via the Quorum

system, then we also require a minimum of six violations in order to deactivate a node.

In order to trick this system, an attacker will need to be selected six times in a row. Otherwise, violations will be cancelled out by the system as other nodes are selected by the quorum system.

Attacker Controlled Masternodes / Total Masternodes	Required Picked Times In A Row	Probability of success $(n/t)^r$	DASH Required
1/2300	6	6.75e-21	1,000DASH
10/2300	6	6.75e-15	10,000DASH
100/2300	6	6.75e-09	100,000DASH
500/2300	6	0 %	500,000DASH
1000/2300	6	1 %	1,000,000DASH

Table 1. The probability of tricking the system representing one individual masternode as failing proof-of-service

Where:

n is the total number of nodes controlled by the attacker

t is the total number of masternodes in the network

r is the depth of the chain

The selection of masternodes is pseudo random based on the Quorum system

2.5 Masternode Protocol

The masternodes are propagated around the network using a series of protocol extensions including a masternode announce message and masternode ping message. These two messages are all that is needed to make a node active on the network, beyond these there are other messages for executing a proof-of-service request, PrivateSend and InstantSend.

Masternodes are originally formed by sending 1,000 DASH to a specific address in a wallet that will “activate” the node making it capable of being propagated across the network. A secondary private key is created that is used for signing all further messages. The latter key allows the wallet to be completely locked when running in a standalone mode.

A cold mode is made possible by utilizing the secondary private key on two separate machines. The primary “hot” client signs the 1,000 DASH input including the secondary signing private key in the message. Soon after the “cold” client sees a message including its secondary key and activates as a masternode. This allows the “hot” client to be deactivated (client turned off) and leaves no possibility of an attacker gaining access to the 1,000 DASH by gaining access to the masternode after activation.

Upon starting, a masternode sends a “Masternode Announce” message to the network, containing:

Message: (1K DASH Input, Reachable IP Address, Signature, Signature Time, 1K Dash Public Key, Secondary Public Key, Donation Public Key, Donation Percentage)

Every 15 minutes thereafter, a ping message is sent proving the node is still alive.

*Message: (1K DASH Input, Signature (using secondary key),
Signature Time, Stop)*

After a time-to-live has expired the network will remove an inactive node from the network, causing the node to not be used by clients or paid. Nodes can also ping the network constantly, but if they do not have their ports open, they will eventually be flagged as inactive and not be paid.

2.6 Propagation of the Masternode List

New clients entering the Dash network must be made aware of the currently active masternodes on the network to be able to utilize their services. As soon as they join the mesh network, a command is sent to their peers asking for the known list of masternodes. A cache object is used for clients to record masternodes and their current status, so when clients restart they will simply load this file rather than asking for the full list of masternodes.

2.7 Payments via Mining and Enforcement

To ensure that each masternode is paid its fair share of the block reward, the network must enforce that blocks pay the correct masternode. If a miner is non-compliant their blocks must be rejected by the network, otherwise cheating will be incentivized.

We propose a strategy where masternodes form quorums, select a winning masternode and broadcast their message. After N messages have been broadcast to select the same target payee, a consensus will be formed and that block in question will be required to pay that masternode.

When mining on the network, pool software (websites that merge the efforts of individual miners) use the RPC API interface to get information about how to make a block. To pay the masternodes, this interface must be extended by adding a secondary payee to GetBlockTemplate. Pools then propagate their successfully mined blocks, with a split payment between themselves and a masternode.

3 PrivateSend

We believe it is important to have a standard trustless implementation for improving the privacy of its users in the reference client that provides a high degree of privacy. Other clients such as Electrum, Android and iOS will also have the same anonymity layer implemented directly and utilize the protocol extensions. This allows users a common experience anonymizing funds using a well understood system.

PrivateSend is an improved and extended version of the CoinJoin. In addition to the core concept of CoinJoin, we employ a series of improvements such as decentralization, strong anonymity by using a chaining approach, denominations and passive ahead-of-time mixing.

The greatest challenge when improving privacy and fungibility of a cryptocurrency is doing it in a way that does not obscure the entire blockchain. In Bitcoin based crypto currencies, one can tell which outputs are unspent and which are not, commonly called UTXO (unspent transaction output). This results in a public ledger that allows any user to act as guarantor of the integrity of transactions.

The Bitcoin protocol is designed to function without the participation of trusted counterparties, and in their absence, it is critical that auditing capabilities remain readily accessible to the users through the public blockchain. Our goal is to improve privacy and fungibility without losing these key elements that we believe make a successful currency.

By having a decentralized mixing service within the currency we gain the ability to keep the currency itself perfectly fungible.

Fungibility is an attribute of money, that dictates that all units of a currency should remain equal. When you receive money within a currency, it should not come with any history from the previous users of the currency or the users should have an easy way to disassociate themselves from that history, thus keeping all coins equal. At the same time, any user should be able to act as an auditor to guarantee the financial integrity of the public ledger without compromising others privacy.

To improve the fungibility and keep the integrity of the public blockchain, we propose using an ahead-of-time decentralized trustless mixing strategy. To be effective at keeping the currency fungible, this service is directly built into the currency, easy to use and safe for the average user.

3.1 Tracing CoinJoin By Amounts

A common strategy in existing Bitcoin implementations of CoinJoin is simply merging transactions together. This exposes the users to various methods of following the the users coins through these joined

transaction.



Figure 2: An example CoinJoin transaction with 2 users [11][12]

In this transaction, 0.05BTC was sent through the mixer. To identify the source of the money, one simply has to add up the values on the right until they match one of the values on the left.

Breaking apart the transaction:

- $0.05 + 0.0499 + 0.0001(\text{fee}) = 0.10\text{BTC}$.
- $0.0499 + 0.05940182 + 0.0001(\text{fee}) = 0.10940182\text{BTC}$.

This gets exponentially more difficult as more users are added to the mixer. However, these sessions can be retroactively de-anonymized at any point in the future.

3.2 Through Linking and Forward Linking

In other proposed implementations of CoinJoin, it is possible that a user anonymizes money then eventually sends change from that transaction to an exchange or other entity that knows the users identity. This breaks the anonymity and allows the entity to walk backwards through that users transactions. We call this type of attack “Forward Linking”:

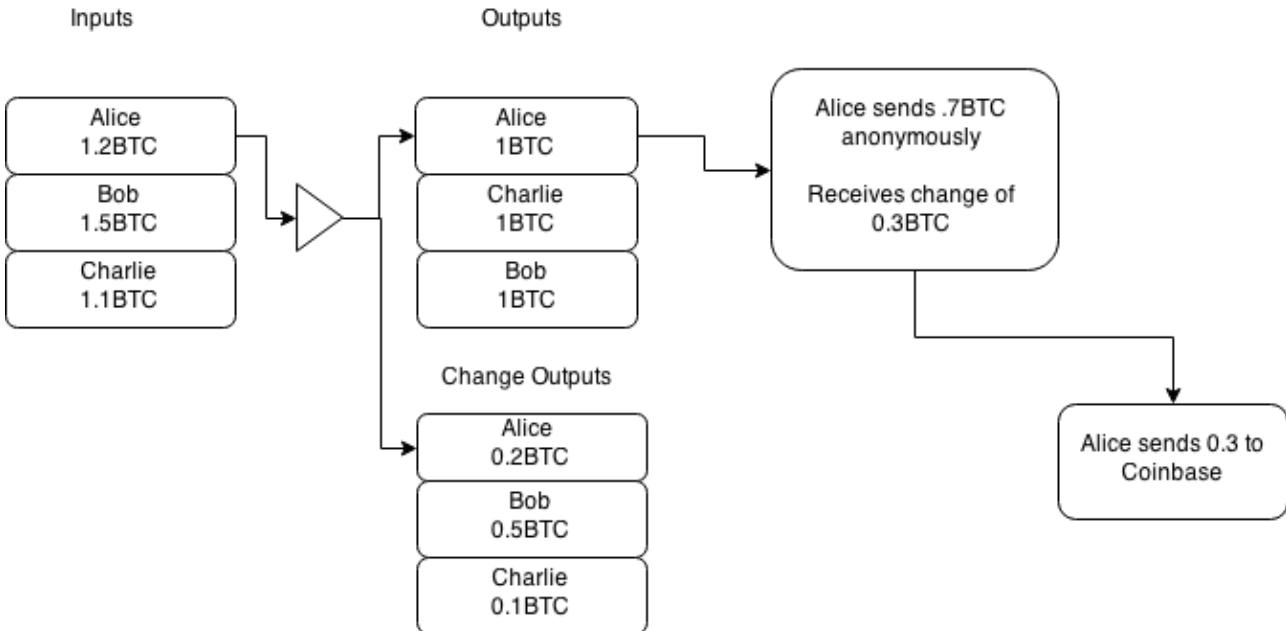


Figure 3: Forward Change Linking

In this example, Alice anonymizes 1.2 BTC, which goes to two outputs, 1 BTC and 0.2 BTC. She then spends 0.7 BTC from the 1 BTC output, receiving change of 0.3 BTC. That 0.3 BTC then goes to an identifiable source, confirming Alice also spent the 0.7 BTC in the prior transaction.

To identify the sender of the anonymous transaction, start at the “exchange” transaction and go backwards in the blockchain till you get to the “Alice sends 0.7 BTC anonymously”. As the exchange, you know it was your user who just recently bought something anonymously, thus breaking the anonymity completely. We call this type of attack “Through Change Linking”.

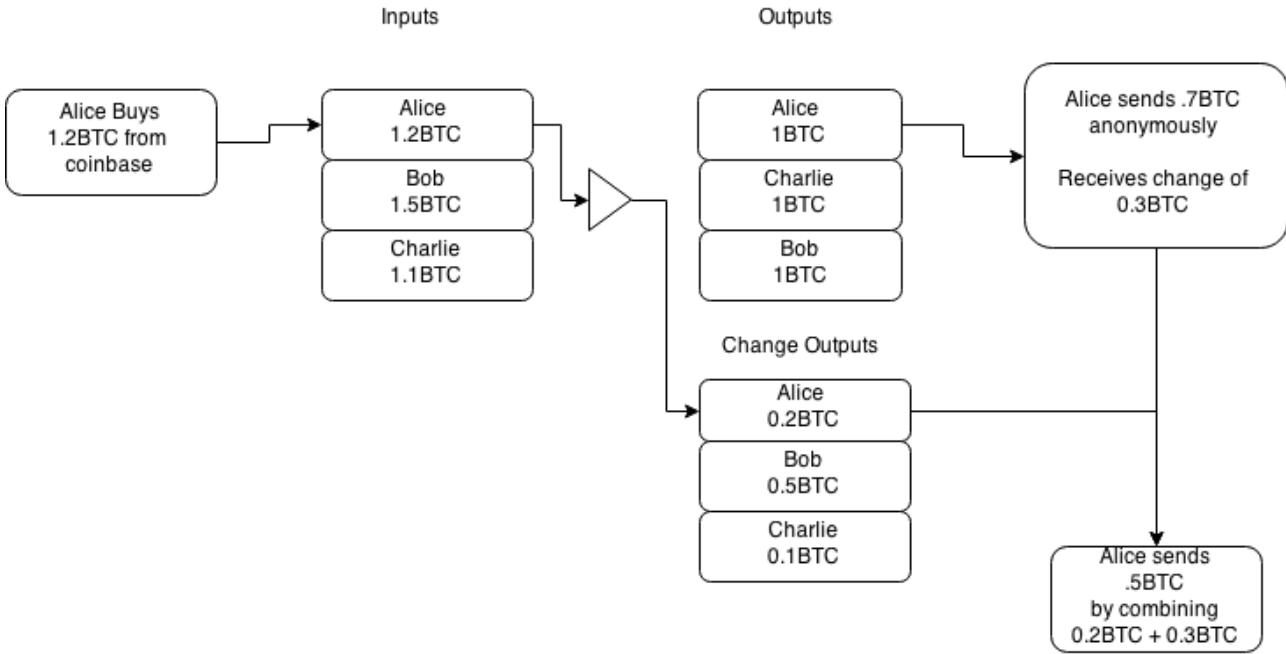


Figure 4: Through Change Linking

In the second example, Alice buys 1.2 BTC from coinbase, then anonymizes this amount into a 1 BTC output. She then spends the 1 BTC, receives change in the amount of 0.3 BTC and then combines that with her 0.2 BTC earlier change.

By combining the change from the anonymous transaction (0.3 BTC) and the change she received from the CoinJoin transaction, you can link the entire history before and after, completely breaking the anonymity.

3.3 Improved Privacy and Denial-of-Service (DOS) Resistance

PrivateSend uses the fact that a transaction can be formed by multiple parties and made out to multiple parties to merge funds together in a way where they cannot be uncoupled thereafter. Given that all PrivateSend transactions are setup for users to pay themselves, the system is highly secure against theft and users

coins always remain safe. Currently, PrivateSend mixing requires at least three participants.

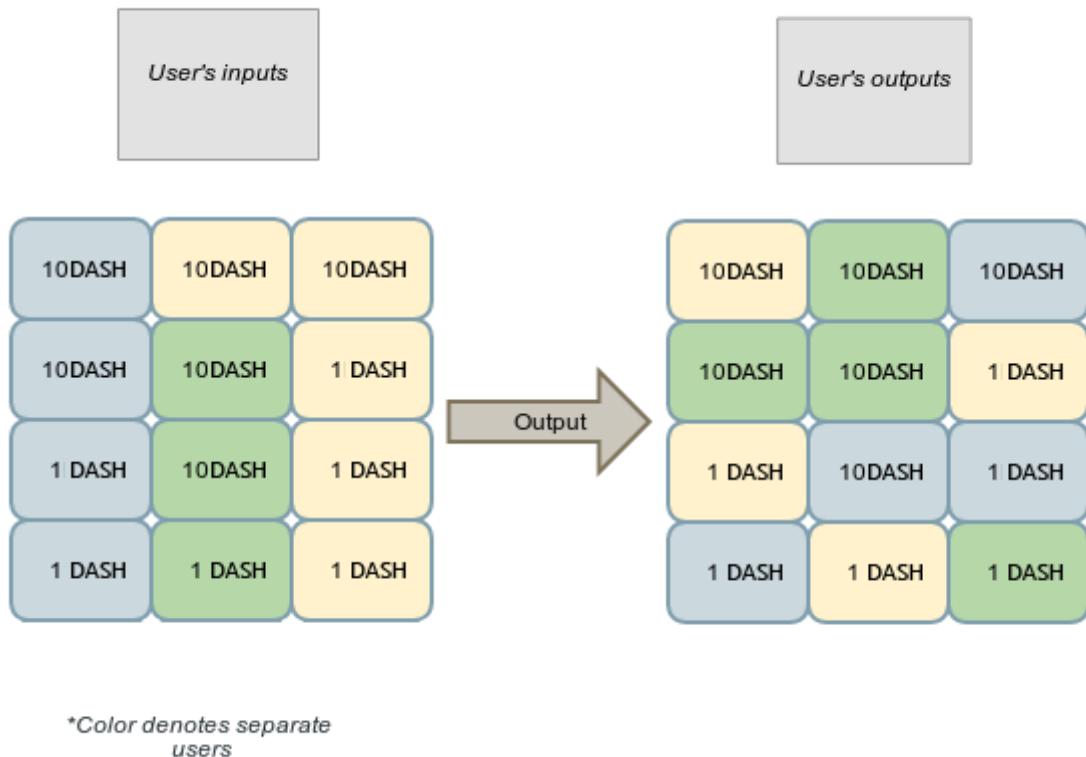


Figure 5: Three users submit denominated funds into a common transaction. Users pay themselves back in the form of new outputs, which are randomly ordered.

To improve the privacy of the system as a whole we propose using common denominations of 0.1DASH, 1DASH, 10DASH AND 100DASH. In each mixing session, all users should submit the same denominations as inputs and outputs. In addition to denominations, fees should be removed from the transactions and charged in bulk in separate, sporadic unlinkable transactions.

To address the possible DOS attacks, we propose all users submit a transaction as collateral to the pool when joining. This transaction will be made out to themselves and will pay a high fee to miners. In

the case when a user submits a request to the mixing pool, they must provide collateral at the beginning of this exchange. If at any point any user fails to cooperate, by refusing to sign for example, the collateral transaction will be broadcast automatically. This will make it expensive to carry out a sustained attack on the privacy network.

3.4 Passive Anonymization of funds and chaining

PrivateSend is limited to 1,000DASH per session and requires multiple sessions to thoroughly anonymize significant amounts of money. To make the user experience easy and timing attacks very difficult, PrivateSend runs in a passive mode. At set intervals, a user's client will request to join with other clients via a masternode. Upon entry into the masternode, a queue object is propagated throughout the network detailing the denominations the user is looking to anonymize, but no information that can be used to identify the user.

Each PrivateSend session can be thought of as an independent event increasing the anonymity of user's funds. However each session is limited to three clients, so an observer has a one in three chance of being able to follow a transaction. To increase the quality of anonymity provided, a chaining approach is employed, which funds are sent through multiple masternodes, one after another.

Depth Of The Chain	Possible Users $(n)^r$
2	9
4	81
8	6561

Table 2. How many users could possibly be involved in N mixing sessions.

3.5 Security Considerations

As transactions are merged, masternodes can possibly “snoop” on users funds as they pass through. This is not considered a serious limitation due to the requirement for masternode’s to hold 1,000DASH and the fact that users utilize random masternodes that they select to host their joins. The probability of following a transaction throughout a chaining event can be calculated as follows:

Attacker Controlled Masternodes / Total Masternodes	Depth Of The Chain	Probability of success $(n / t)^r$	DASH Required
10/1010	2	9.80e-05	10,000DA SH
10/1010	4	9.60e-09	10,000DA SH
10/1010	8	9.51e-11	10,000DA SH
100/1100	2	8.26e-03	100,000DASH
100/1100	4	6.83e-05	100,000DASH

100/1100	8	4.66e-09	100,000D ASH
1000/2000	2	25 %	1,000,000 DASH
1000/2000	4	6 %	1,000,000 DASH
1000/2000	8	0 %	1,000,000 DASH
2000/3000	2	44 %	2,000,000 DASH
2000/3000	4	20 %	2,000,000 DASH
2000/3000	8	4 %	2,000,000 DASH

Table 3. The probability of follow a PrivateSend transaction on the network given the attacker controls N Nodes.

Where:

n is the total number of nodes controlled by the attacker

t is the total number of masternodes in the network

r is the depth of the chain

The selection of masternodes is random.

Considering the limited supply of DASH (8.2 million at the time of writing, August 2018) and the low liquidity available on the market, it becomes an impossibility to attain a large enough number of masternodes to succeed at such an attack.

Extending the system by blinding masternodes to the transactions taking place on their node will also greatly enhance the security of the system.

3.6 Masternode Blinding via Relay System

In Section 3.4 we describe the probabilities of following a single transaction through multiple sessions of PrivateSend mixing. This can further be addressed by blinding masternodes, so they cannot see which inputs/outputs belong to which users. To do this we propose a simple relay system that users can use to protect their identity.

Instead of a user submitting the inputs and outputs directly into the pool, they will pick a random masternode from the network and request that it relays the inputs/outputs/signatures to the target masternode. This means that the masternode will receive N sets of inputs/outputs and N sets of signatures. Each set belongs to one of the users, but the masternode can't know which belongs to which.

4 Instant Transactions via InstantSend

By utilizing masternode quorums, users are able to send and receive instant irreversible transactions. Once a quorum has been formed, the inputs of the transaction are locked to only be spendable in a specific transaction, a transaction lock takes about four seconds to be set currently on the network. If consensus is reached on a lock by the masternode network, all conflicting transactions or conflicting blocks would be rejected thereafter, unless they matched the exact transaction ID of the lock in place.

This will allow vendors to use mobile devices in place of traditional POS systems for real world commerce and users to quickly settle face-to-face non commercial transactions as with traditional cash.

This is done without a central authority. An extensive overview of this feature can be found in the InstantSend white paper [9].

5 Additional Improvements

5.1 X11 hashing algorithm

X11 is a widely used hashing algorithm, which takes a different approach, known as algorithm chaining. X11 consists of all 11 SHA3 contestants [13], each hash is calculated then submitted to the next algorithm in the chain. By utilizing multiple algorithms, the likelihood that an ASIC is created for the currency is minimal until a later part of its life cycle.

In the life cycle of Bitcoin, mining began with hobbyists which used Central Processing Units (CPUs) to mine the currency, then shortly after Graphics Processing Units (GPUs) software was created, which quickly replaced the CPUs. Years after the GPUs cycle, ASICs or Application Specific Integrated Circuits were created, which quickly replaced the GPUs.

Due to the complexity and die size required to create an ASIC to mine X11, we expect that it will take considerably longer than it did in Bitcoin, allowing for hobbyists to take part in the mining for a longer period of time. We believe this is highly important for good distribution and growth of a cryptocurrency.

Another benefit of the chaining hashing approach is high end CPUs give an average return similar to that of GPUs. Also GPUs have

been reported to run 30-50% cooler, with less wattage than the Scrypt algorithm used by most current cryptocurrencies.

5.2 Mining Supply

A different approach to restricting the inflation of mining is taken in Dash, using a 7% reduction of the supply per year. This is done as opposed to halving implemented by other currencies. In addition supply each block is directly tied to the amount of miners on the network; more miners result in lower mining rewards.

Production of Dash is scheduled to carry on throughout this century and onto the next, slowly grinding down until finally near the year 2150, production will cease.

Dash Currency Supply and Mining Reward Schedule

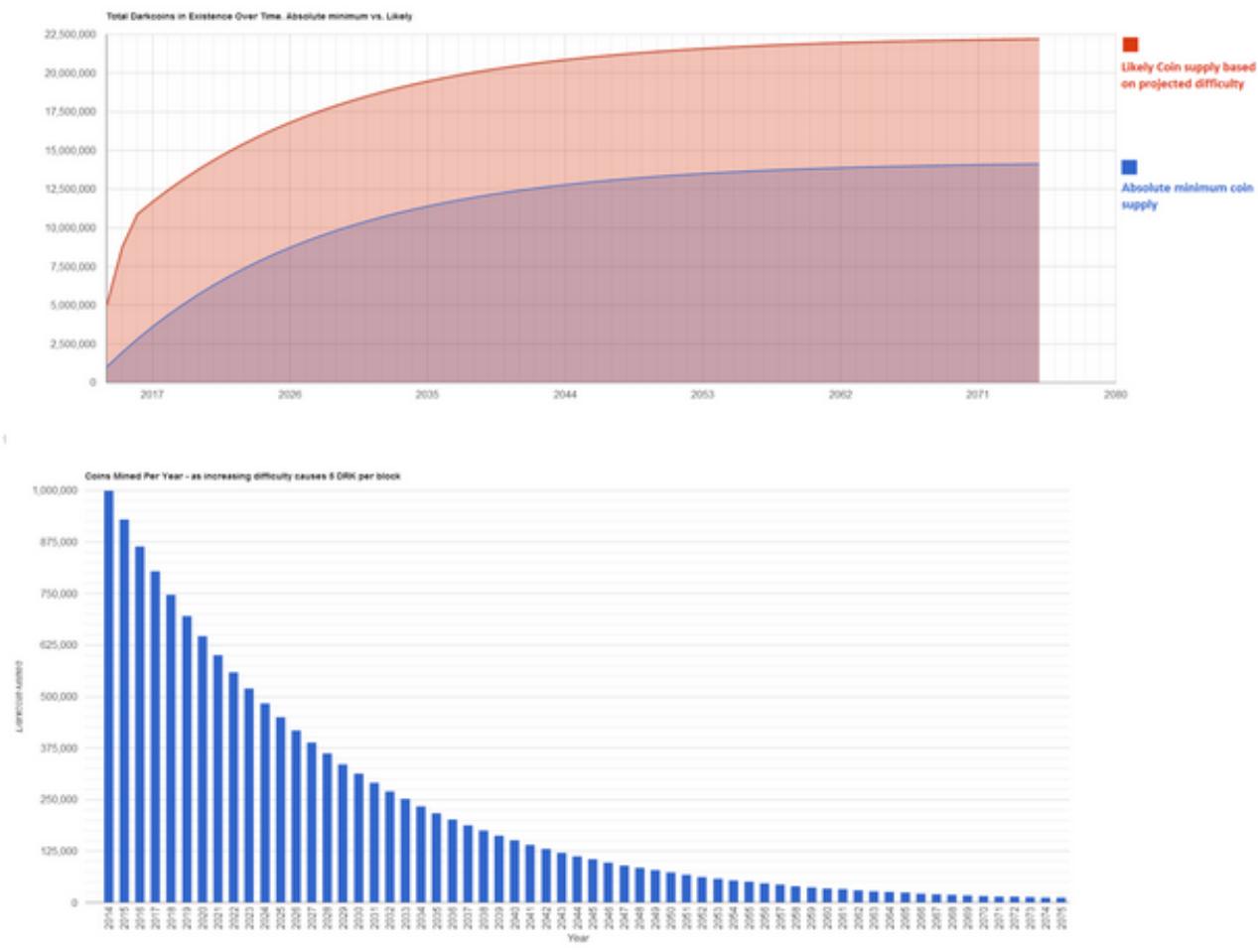


Figure 6: Mining Reward Schedule

6 Conclusion

This paper introduces various concepts to improve the design of bitcoin resulting in improved privacy and fungibility for the average user, less price volatility and quicker message propagation throughout the network. This is all accomplished by utilizing an incentivized two-tier model, rather than the existing single-tier model in other cryptocurrencies such as Bitcoin. By utilizing this alternative network design it becomes possible to add many types of services such as decentralized mixing of coins, instant transactions and decentralized oracles using masternode quorums.

1. A Next-Generation Smart Contract and Decentralized Application Platform

[![Documentation](<https://img.shields.io/badge/gitter-Docs%20chat-4AB495.svg>)](<https://gitter.im/ethereum/documentation>)

> An introductory paper to Ethereum, introduced before launch, which is maintained.

Satoshi Nakamoto's development of Bitcoin in 2009 has often been hailed as a radical development in money and currency, being the first example of a digital asset which simultaneously has no backing or [intrinsic](<http://bitcoinmagazine.com/8640/an-exploration-of-intrinsic-value-what-it-is-why-bitcoin-doesnt-have-it-and-why-bitcoin-does-have-it/>) and no centralized issuer or controller.

However, another - arguably more important - part of the Bitcoin experiment is the underlying blockchain technology as a tool of distributed consensus, and attention is rapidly starting to shift to this other aspect of Bitcoin. Commonly cited alternative applications of blockchain technology include using on-blockchain digital assets to represent custom currencies and financial instruments ([colored] (https://docs.google.com/a/buterin.com/document/d/1AnkP_cVZTCMLIzw4DvsW6M8Q2JC0IlzrTLuoWu2z1BE/edit)), the ownership of an underlying physical device ([smart](https://en.bitcoin.it/wiki/Smart_Property)), non-fungible assets such as domain names ([Namecoin](<http://namecoin.org>)), as well as more complex applications involving having digital assets being directly controlled by a piece of code implementing arbitrary rules ([smart] (<http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/idea.html>)) or even blockchain-based [decentralized] (<http://bitcoinmagazine.com/7050/bootstrapping-a-decentralized-autonomous-corporation-part-i/>) (DAOs). What Ethereum intends to provide is a blockchain with a built-in fully fledged Turing-complete programming language that can be used to create "contracts" that can be used to encode arbitrary state transition functions, allowing users to create any of the systems described above, as well as many others that we have not yet imagined, simply by writing up the logic in a few lines of code.

- Contents**

- [Introduction](#introduction-to-bitcoin-and-existing-concepts)
 - [History](#history)
 - [Bitcoin As A State Transition System](#bitcoin-as-a-state-transition-system)
 - [Mining](#mining)
 - [Merkle Trees](#merkle-trees)
 - [Alternative Blockchain Applications](#alternative-blockchain-applications)
 - [Scripting](#scripting)
- [Ethereum](#ethereum)
 - [Philosophy](#philosophy)
 - [Ethereum Accounts](#ethereum-accounts)
 - [Messages and Transactions](#messages-and-transactions)
 - [Messages](#messages)
 - [Ethereum State Transition Function](#ethereum-state-transition-function)
 - [Code Execution](#code-execution)
 - [Blockchain and Mining](#blockchain-and-mining)
- [Applications](#applications)
 - [Token Systems](#token-systems)
 - [Financial derivatives and Stable-Value Currencies](#financial-derivatives-and-stable-value-currencies)
 - [Identity and Reputation Systems](#identity-and-reputation-systems)
 - [Decentralized File Storage](#decentralized-file-storage)
 - [Decentralized Autonomous Organizations](#decentralized-autonomous-organizations)
 - [Further Applications](#further-applications)
- [Miscellanea](#miscellanea-and-concerns)
 - [Modified GHOST Implementation](#modified-ghost-implementation)
 - [Fees](#fees)

- [Computation And Turing-Completeness] (#computation-and-turing-completeness)
 - [Currency And Issuance] (#currency-and-issuance)
 - [Mining Centralization] (#mining-centralization)
 - [Scalability] (#scalability)
- [Conclusion](#conclusion) - [Notes](#notes-and-further-reading)

- [Notes] (#notes)
- [Further Reading] (#further-reading)

1. Introduction to Bitcoin and Existing Concepts

i. History

The concept of decentralized digital currency, as well as alternative applications like property registries, has been around for decades. The anonymous e-cash protocols of the 1980s and the 1990s, mostly reliant on a cryptographic primitive known as Chaumian blinding, provided a currency with a high degree of privacy, but the protocols largely failed to gain traction because of their reliance on a centralized intermediary. In 1998, Wei Dai's [b-money](<http://www.weidai.com/bmoney.txt>) became the first proposal to introduce the idea of creating money through solving computational puzzles as well as decentralized consensus, but the proposal was scant on details as to how decentralized consensus could actually be implemented. In 2005, Hal Finney introduced a concept of [reusable](<http://nakamotoinstitute.org/finney/rpow/>), a system which uses ideas from b-money together with Adam Back's computationally difficult Hashcash puzzles to create a concept for a cryptocurrency, but once again fell short of the ideal by relying on trusted computing as a backend. In 2009, a decentralized currency was for the first time implemented in practice by Satoshi Nakamoto, combining established primitives for managing ownership through public key cryptography with a consensus algorithm for keeping track of who owns coins, known as "proof of work".

The mechanism behind proof of work was a breakthrough in the space because it simultaneously solved two problems. First, it provided a simple and moderately effective consensus algorithm, allowing nodes in the network to collectively agree on a set of canonical updates to the state of the Bitcoin ledger. Second, it provided a mechanism for allowing free entry into the consensus

process, solving the political problem of deciding who gets to influence the consensus, while simultaneously preventing sybil attacks. It does this by substituting a formal barrier to participation, such as the requirement to be registered as a unique entity on a particular list, with an economic barrier - the weight of a single node in the consensus voting process is directly proportional to the computing power that the node brings. Since then, an alternative approach has been proposed called proof of stake, calculating the weight of a node as being proportional to its currency holdings and not computational resources; the discussion of the relative merits of the two approaches is beyond the scope of this paper but it should be noted that both approaches can be used to serve as the backbone of a cryptocurrency.

Here is a blog post from Vitalik Buterin, the founder of Ethereum, on [Ethereum](<https://vitalik.ca/2017-09-15-prehistory.html>). [Here](<https://blog.ethereum.org/2016/02/09/cut-and-try-building-a-dream/>) is another blog post with more history.

i. Bitcoin As A State Transition System

![statetransition.png](<https://raw.githubusercontent.com/ethereumbuilders/GitBook/master/en/vitalik-diagrams/statetransition.png>)

From a technical standpoint, the ledger of a cryptocurrency such as Bitcoin can be thought of as a state transition system, where there is a "state" consisting of the ownership status of all existing bitcoins and a "state transition function" that takes a state and a transaction and outputs a new state which is the result. In a standard banking system, for example, the state is a balance sheet, a transaction is a request to move \$X from A to B, and the state transition function reduces the value in A's account by \$X and increases the value in B's account by \$X. If A's account has less than \$X in the first place, the state transition function returns an error. Hence, one can formally define:

APPLY(S, TX) → S' or ERROR

In the banking system defined above:

```
APPLY({ Alice: $50, Bob: $50 }, "send $20 from Alice to Bob") = { Alice: $30, Bob: $70 }
```

But:

```
APPLY({ Alice: $50, Bob: $50 }, "send $70 from Alice to Bob") = ERROR
```

The "state" in Bitcoin is the collection of all coins (technically, "unspent transaction outputs" or UTXO) that have been mined and not yet spent, with each UTXO having a denomination and an owner (defined by a 20-byte address which is essentially a cryptographic public key^[fn.](<https://github.com/ethereum/wiki/wiki/White-Paper#notes>)). A transaction contains one or more inputs, with each input containing a reference to an existing UTXO and a cryptographic signature produced by the private key associated with the owner's address, and one or more outputs, with each output containing a new UTXO to be added to the state.

The state transition function `APPLY(S, TX) -> S` can be defined roughly as follows:

1. For each input in `TX`:

- * If the referenced UTXO is not in `S`, return an error.
- * If the provided signature does not match the owner of the UTXO, return an error.

2. If the sum of the denominations of all input UTXO is less than the sum of the denominations of all output UTXO, return an error.
3. Return `S` with all input UTXO removed and all output UTXO added.

The first half of the first step prevents transaction senders from spending coins that do not exist, the second half of the first step prevents transaction senders from spending other people's coins, and the second step enforces conservation of value. In order to use this for payment, the protocol is as follows. Suppose Alice wants to send 11.7 BTC to Bob. First, Alice will look for a set of available UTXO that she owns that totals up to at least 11.7 BTC. Realistically, Alice will not be able to get exactly 11.7 BTC; say that

the smallest she can get is $6+4+2=12$. She then creates a transaction with those three inputs and two outputs. The first output will be 11.7 BTC with Bob's address as its owner, and the second output will be the remaining 0.3 BTC "change", with the owner being Alice herself.

i. Mining

![block_picture.jpg]([https://raw.githubusercontent.com/ethereumbuilders/GitBook/master/en/vitalik-diagrams/block.png\)](https://raw.githubusercontent.com/ethereumbuilders/GitBook/master/en/vitalik-diagrams/block.png)

If we had access to a trustworthy centralized service, this system would be trivial to implement; it could simply be coded exactly as described, using a centralized server's hard drive to keep track of the state. However, with Bitcoin we are trying to build a decentralized currency system, so we will need to combine the state transition system with a consensus system in order to ensure that everyone agrees on the order of transactions. Bitcoin's decentralized consensus process requires nodes in the network to continuously attempt to produce packages of transactions called "blocks". The network is intended to produce roughly one block every ten minutes, with each block containing a timestamp, a nonce, a reference to (ie. hash of) the previous block and a list of all of the transactions that have taken place since the previous block. Over time, this creates a persistent, ever-growing, "blockchain" that constantly updates to represent the latest state of the Bitcoin ledger.

The algorithm for checking if a block is valid, expressed in this paradigm, is as follows:

1. Check if the previous block referenced by the block exists and is valid.
2. Check that the timestamp of the block is greater than that of the previous block[fn.](<https://github.com/ethereum/wiki/wiki/White-Paper#notes>) and less than 2 hours into the future
3. Check that the proof of work on the block is valid.
4. Let `S[0]` be the state at the end of the previous block.
5. Suppose `TX` is the block's transaction list with `n` transactions. For all `i` in `0...n-1`, set `S[i+1] = APPLY(S[i], TX[i])` If any application returns an error, exit and return false.
6. Return true, and register `S[n]` as the state at the end of this block.

Essentially, each transaction in the block must provide a valid state transition from what was the canonical state before the transaction was executed to some new state. Note that the state is not encoded in the block in any way; it is purely an abstraction to be remembered by the validating node and can only be (securely) computed for any block by starting from the genesis state and sequentially applying every transaction in every block. Additionally, note that the order in which the miner includes transactions into the block matters; if there are two transactions A and B in a block such that B spends a UTXO created by A, then the block will be valid if A comes before B but not otherwise.

The one validity condition present in the above list that is not found in other systems is the requirement for "proof of work". The precise condition is that the double-SHA256 hash of every block, treated as a 256-bit number, must be less than a dynamically adjusted target, which as of the time of this writing is approximately 2^{187} . The purpose of this is to make block creation computationally "hard", thereby preventing sybil attackers from remaking the entire blockchain in their favor. Because SHA256 is designed to be a completely unpredictable pseudorandom function, the only way to create a valid block is simply trial and error, repeatedly incrementing the nonce and seeing if the new hash matches.

At the current target of $\sim 2^{187}$, the network must make an average of $\sim 2^{69}$ tries before a valid block is found; in general, the target is recalibrated by the network every 2016 blocks so that on average a new block is produced by some node in the network every ten minutes. In order to compensate miners for this computational work, the miner of every block is entitled to include a transaction giving themselves 12.5 BTC out of nowhere. Additionally, if any transaction has a higher total denomination in its inputs than in its outputs, the difference also goes to the miner as a "transaction fee". Incidentally, this is also the only mechanism by which BTC are issued; the genesis state contained no coins at all.

In order to better understand the purpose of mining, let us examine what happens in the event of a malicious attacker. Since Bitcoin's

underlying cryptography is known to be secure, the attacker will target the one part of the Bitcoin system that is not protected by cryptography directly: the order of transactions. The attacker's strategy is simple:

1. Send 100 BTC to a merchant in exchange for some product (preferably a rapid-delivery digital good)
2. Wait for the delivery of the product
3. Produce another transaction sending the same 100 BTC to himself
4. Try to convince the network that his transaction to himself was the one that came first.

Once step (1) has taken place, after a few minutes some miner will include the transaction in a block, say block number 270. After about one hour, five more blocks will have been added to the chain after that block, with each of those blocks indirectly pointing to the transaction and thus "confirming" it. At this point, the merchant will accept the payment as finalized and deliver the product; since we are assuming this is a digital good, delivery is instant. Now, the attacker creates another transaction sending the 100 BTC to himself. If the attacker simply releases it into the wild, the transaction will not be processed; miners will attempt to run `APPLY(S,TX)` and notice that `TX` consumes a UTXO which is no longer in the state. So instead, the attacker creates a "fork" of the blockchain, starting by mining another version of block 270 pointing to the same block 269 as a parent but with the new transaction in place of the old one. Because the block data is different, this requires redoing the proof of work. Furthermore, the attacker's new version of block 270 has a different hash, so the original blocks 271 to 275 do not "point" to it; thus, the original chain and the attacker's new chain are completely separate. The rule is that in a fork the longest blockchain is taken to be the truth, and so legitimate miners will work on the 275 chain while the attacker alone is working on the 270 chain. In order for the attacker to make his blockchain the longest, he would need to have more computational power than the rest of the network combined in order to catch up (hence, "51% attack").

i. Merkle Trees

![SPV](https://raw.githubusercontent.com/ethereum/www/master-postsale/src/extras/gh_wiki/spv_bitcoin.png)

Left: it suffices to present only a small number of nodes in a Merkle tree to give a proof of the validity of a branch.

Right: any attempt to change any part of the Merkle tree will eventually lead to an inconsistency somewhere up the chain.

An important scalability feature of Bitcoin is that the block is stored in a multi-level data structure. The "hash" of a block is actually only the hash of the block header, a roughly 200-byte piece of data that contains the timestamp, nonce, previous block hash and the root hash of a data structure called the Merkle tree storing all transactions in the block. A Merkle tree is a type of binary tree, composed of a set of nodes with a large number of leaf nodes at the bottom of the tree containing the underlying data, a set of intermediate nodes where each node is the hash of its two children, and finally a single root node, also formed from the hash of its two children, representing the "top" of the tree. The purpose of the Merkle tree is to allow the data in a block to be delivered piecemeal: a node can download only the header of a block from one source, the small part of the tree relevant to them from another source, and still be assured that all of the data is correct. The reason why this works is that hashes propagate upward: if a malicious user attempts to swap in a fake transaction into the bottom of a Merkle tree, this change will cause a change in the node above, and then a change in the node above that, finally changing the root of the tree and therefore the hash of the block, causing the protocol to register it as a completely different block (almost certainly with an invalid proof of work).

The Merkle tree protocol is arguably essential to long-term sustainability. A "full node" in the Bitcoin network, one that stores and processes the entirety of every block, takes up about 15 GB of disk space in the Bitcoin network as of April 2014, and is growing by over a gigabyte per month. Currently, this is viable for some desktop computers and not phones, and later on in the future only businesses and hobbyists will be able to participate. A protocol

known as "simplified payment verification" (SPV) allows for another class of nodes to exist, called "light nodes", which download the block headers, verify the proof of work on the block headers, and then download only the "branches" associated with transactions that are relevant to them. This allows light nodes to determine with a strong guarantee of security what the status of any Bitcoin transaction, and their current balance, is while downloading only a very small portion of the entire blockchain.

i. Alternative Blockchain Applications

The idea of taking the underlying blockchain idea and applying it to other concepts also has a long history. In 1998, Nick Szabo came out with the concept of [secure](<http://nakamotoinstitute.org/secure-property-titles/>), a document describing how "new advances in replicated database technology" will allow for a blockchain-based system for storing a registry of who owns what land, creating an elaborate framework including concepts such as homesteading, adverse possession and Georgian land tax. However, there was unfortunately no effective replicated database system available at the time, and so the protocol was never implemented in practice. After 2009, however, once Bitcoin's decentralized consensus was developed a number of alternative applications rapidly began to emerge.

- **Namecoin** - created in 2010, [Namecoin](<https://namecoin.org/>) is best described as a decentralized name registration database. In decentralized protocols like Tor, Bitcoin and BitMessage, there needs to be some way of identifying accounts so that other people can interact with them, but in all existing solutions the only kind of identifier available is a pseudorandom hash like `1LW79wp5ZBqaHW1jL5TCiBCrhQYtHagUWy`. Ideally, one would like to be able to have an account with a name like "george". However, the problem is that if one person can create an account named "george" then someone else can use the same process to register "george" for themselves as well and impersonate them. The only solution is a first-to-file paradigm, where the first registerer succeeds and the second fails - a problem perfectly suited for the Bitcoin consensus

protocol. Namecoin is the oldest, and most successful, implementation of a name registration system using such an idea.

- **Colored coins** - the purpose of [colored](https://docs.google.com/a/buterin.com/document/d/1AnkP_cVZTCMLIzw4DvsW6M8Q2JC0IlzrTLuoWu2z1BE/edit) is to serve as a protocol to allow people to create their own digital currencies - or, in the important trivial case of a currency with one unit, digital tokens, on the Bitcoin blockchain. In the colored coins protocol, one "issues" a new currency by publicly assigning a color to a specific Bitcoin UTXO, and the protocol recursively defines the color of other UTXO to be the same as the color of the inputs that the transaction creating them spent (some special rules apply in the case of mixed-color inputs). This allows users to maintain wallets containing only UTXO of a specific color and send them around much like regular bitcoins, backtracking through the blockchain to determine the color of any UTXO that they receive.
- **Metacoins** - the idea behind a metacoin is to have a protocol that lives on top of Bitcoin, using Bitcoin transactions to store metacoin transactions but having a different state transition function, `APPLY`. Because the metacoin protocol cannot prevent invalid metacoin transactions from appearing in the Bitcoin blockchain, a rule is added that if `APPLY'(S,TX)` returns an error, the protocol defaults to `APPLY'(S,TX) = S`. This provides an easy mechanism for creating an arbitrary cryptocurrency protocol, potentially with advanced features that cannot be implemented inside of Bitcoin itself, but with a very low development cost since the complexities of mining and networking are already handled by the Bitcoin protocol. Metacoins have been used to implement some classes of financial contracts, name registration and decentralized exchange.

Thus, in general, there are two approaches toward building a consensus protocol: building an independent network, and building a protocol on top of Bitcoin. The former approach, while reasonably successful in the case of applications like Namecoin, is difficult to

implement; each individual implementation needs to bootstrap an independent blockchain, as well as building and testing all of the necessary state transition and networking code. Additionally, we predict that the set of applications for decentralized consensus technology will follow a power law distribution where the vast majority of applications would be too small to warrant their own blockchain, and we note that there exist large classes of decentralized applications, particularly decentralized autonomous organizations, that need to interact with each other.

The Bitcoin-based approach, on the other hand, has the flaw that it does not inherit the simplified payment verification features of Bitcoin. SPV works for Bitcoin because it can use blockchain depth as a proxy for validity; at some point, once the ancestors of a transaction go far enough back, it is safe to say that they were legitimately part of the state. Blockchain-based meta-protocols, on the other hand, cannot force the blockchain not to include transactions that are not valid within the context of their own protocols. Hence, a fully secure SPV meta-protocol implementation would need to backward scan all the way to the beginning of the Bitcoin blockchain to determine whether or not certain transactions are valid. Currently, all "light" implementations of Bitcoin-based meta-protocols rely on a trusted server to provide the data, arguably a highly suboptimal result especially when one of the primary purposes of a cryptocurrency is to eliminate the need for trust.

i. Scripting

Even without any extensions, the Bitcoin protocol actually does facilitate a weak version of a concept of "smart contracts". UTXO in Bitcoin can be owned not just by a public key, but also by a more complicated script expressed in a simple stack-based programming language. In this paradigm, a transaction spending that UTXO must provide data that satisfies the script. Indeed, even the basic public key ownership mechanism is implemented via a script: the script takes an elliptic curve signature as input, verifies it against the transaction and the address that owns the UTXO, and returns 1 if the verification is successful and 0 otherwise. Other, more complicated, scripts exist for various additional use cases. For example, one can construct a script that requires signatures from

two out of a given three private keys to validate ("multisig"), a setup useful for corporate accounts, secure savings accounts and some merchant escrow situations. Scripts can also be used to pay bounties for solutions to computational problems, and one can even construct a script that says something like "this Bitcoin UTXO is yours if you can provide an SPV proof that you sent a Dogecoin transaction of this denomination to me", essentially allowing decentralized cross-cryptocurrency exchange.

However, the scripting language as implemented in Bitcoin has several important limitations:

- ****Lack of Turing-completeness**** - that is to say, while there is a large subset of computation that the Bitcoin scripting language supports, it does not nearly support everything. The main category that is missing is loops. This is done to avoid infinite loops during transaction verification; theoretically it is a surmountable obstacle for script programmers, since any loop can be simulated by simply repeating the underlying code many times with an if statement, but it does lead to scripts that are very space-inefficient. For example, implementing an alternative elliptic curve signature algorithm would likely require 256 repeated multiplication rounds all individually included in the code.
- ****Value-blindness**** - there is no way for a UTXO script to provide fine-grained control over the amount that can be withdrawn. For example, one powerful use case of an oracle contract would be a hedging contract, where A and B put in \$1000 worth of BTC and after 30 days the script sends \$1000 worth of BTC to A and the rest to B. This would require an oracle to determine the value of 1 BTC in USD, but even then it is a massive improvement in terms of trust and infrastructure requirement over the fully centralized solutions that are available now. However, because UTXO are all-or-nothing, the only way to achieve this is through the very inefficient hack of having many UTXO of varying denominations (eg. one UTXO of 2^k for every k up to 30) and having O pick which UTXO to send to A and which to B.

- **Lack of state** - a [UTXO](<https://bitcoin.org/en/glossary/unspent-transaction-output>); there is no opportunity for multi-stage contracts or scripts which keep any other internal state beyond that. This makes it hard to make multi-stage options contracts, decentralized exchange offers or two-stage cryptographic commitment protocols (necessary for secure computational bounties). It also means that UTXO can only be used to build simple, one-off contracts and not more complex "stateful" contracts such as decentralized organizations, and makes meta-protocols difficult to implement. Binary state combined with value-blindness also mean that another important application, withdrawal limits, is impossible.
- **Blockchain-blindness** - UTXO are blind to blockchain data such as the nonce, the timestamp and previous block hash. This severely limits applications in gambling, and several other categories, by depriving the scripting language of a potentially valuable source of randomness.

Thus, we see three approaches to building advanced applications on top of cryptocurrency: building a new blockchain, using scripting on top of Bitcoin, and building a meta-protocol on top of Bitcoin. Building a new blockchain allows for unlimited freedom in building a feature set, but at the cost of development time, bootstrapping effort and security. Using scripting is easy to implement and standardize, but is very limited in its capabilities, and meta-protocols, while easy, suffer from faults in scalability. With Ethereum, we intend to build an alternative framework that provides even larger gains in ease of development as well as even stronger light client properties, while at the same time allowing applications to share an economic environment and blockchain security.

1. Ethereum

The intent of Ethereum is to create an alternative protocol for building decentralized applications, providing a different set of tradeoffs that we believe will be very useful for a large class of decentralized applications, with particular emphasis on situations where rapid development time, security for small and rarely used applications, and the ability of different applications to very efficiently interact, are important. Ethereum does this by building what is essentially the ultimate abstract foundational layer: a

blockchain with a built-in Turing-complete programming language, allowing anyone to write smart contracts and decentralized applications where they can create their own arbitrary rules for ownership, transaction formats and state transition functions. A bare-bones version of Namecoin can be written in two lines of code, and other protocols like currencies and reputation systems can be built in under twenty. Smart contracts, cryptographic "boxes" that contain value and only unlock it if certain conditions are met, can also be built on top of the platform, with vastly more power than that offered by Bitcoin scripting because of the added powers of Turing-completeness, value-awareness, blockchain-awareness and state.

i. Philosophy

The design behind Ethereum is intended to follow the following principles:

1. ****Simplicity****: the Ethereum protocol should be as simple as possible, even at the cost of some data storage or time inefficiency.
[fn.](<https://github.com/ethereum/wiki/wiki/White-Paper#notes>) An average programmer should ideally be able to follow and implement the entire specification,[fn.](<https://github.com/ethereum/wiki/wiki/White-Paper#notes>) so as to fully realize the unprecedented democratizing potential that cryptocurrency brings and further the vision of Ethereum as a protocol that is open to all. Any optimization which adds complexity should not be included unless that optimization provides very substantial benefit.
2. ****Universality****: a fundamental part of Ethereum's design philosophy is that Ethereum does not have "features".[fn.](<https://github.com/ethereum/wiki/wiki/White-Paper#notes>) Instead, Ethereum provides an internal Turing-complete scripting language, which a programmer can use to construct any smart contract or transaction type that can be mathematically defined. Want to invent your own financial derivative? With Ethereum, you can. Want to make your own currency? Set it up as an Ethereum contract. Want to set up a full-scale Daemon or Skynet? You may need to have a few thousand interlocking contracts, and be sure to feed them generously, to do that, but nothing is stopping you with Ethereum at your fingertips.
3. ****Modularity****: the parts of the Ethereum protocol should be designed to be as modular and separable as possible. Over the course of development, our goal is to create a program where if one

was to make a small protocol modification in one place, the application stack would continue to function without any further modification. Innovations such as Ethash (see the [Yellow](<https://ethereum.github.io/yellowpaper/paper.pdf#appendix.J>) or [wiki](<https://github.com/ethereum/wiki/wiki/Ethash>), modified Patricia trees ([Yellow](<https://ethereum.github.io/yellowpaper/paper.pdf#appendix.D>), [wiki](<https://github.com/ethereum/wiki/wiki/%5BEnglish%5D-Patricia-Tree>)) and RLP ([YP](<https://ethereum.github.io/yellowpaper/paper.pdf#appendix.B>), [wiki](<https://github.com/ethereum/wiki/wiki/%5BEnglish%5D-RLP>)) should be, and are, implemented as separate, feature-complete libraries. This is so that even though they are used in Ethereum, even if Ethereum does not require certain features, such features are still usable in other protocols as well. Ethereum development should be maximally done so as to benefit the entire cryptocurrency ecosystem, not just itself.

4. **Agility**: details of the Ethereum protocol are not set in stone. Although we will be extremely judicious about making modifications to high-level constructs, for instance with the [sharding](<https://ethresear.ch/t/sharding-phase-1-spec/1407/>), abstracting execution, with only data availability enshrined in consensus. Computational tests later on in the development process may lead us to discover that certain modifications, e.g. to the protocol architecture or to the Ethereum Virtual Machine (EVM), will substantially improve scalability or security. If any such opportunities are found, we will exploit them.

5. **Non-discrimination** and **non-censorship**: the protocol should not attempt to actively restrict or prevent specific categories of usage. All regulatory mechanisms in the protocol should be designed to directly regulate the harm and not attempt to oppose specific undesirable applications. A programmer can even run an infinite loop script on top of Ethereum for as long as they are willing to keep paying the per-computational-step transaction fee.

i. Ethereum Accounts

In Ethereum, the state is made up of objects called "accounts", with each account having a 20-byte address and state transitions being direct transfers of value and information between accounts. An Ethereum account contains four fields:

- The **nonce**, a counter used to make sure each transaction can only be processed once
- The account's current **ether balance**
- The account's **contract code**, if present
- The account's **storage** (empty by default)

"Ether" is the main internal crypto-fuel of Ethereum, and is used to pay transaction fees. In general, there are two types of accounts: **externally owned accounts**, controlled by private keys, and **contract accounts**, controlled by their contract code. An externally owned account has no code, and one can send messages from an externally owned account by creating and signing a transaction; in a contract account, every time the contract account receives a message its code activates, allowing it to read and write to internal storage and send other messages or create contracts in turn.

Note that "contracts" in Ethereum should not be seen as something that should be "fulfilled" or "complied with"; rather, they are more like "autonomous agents" that live inside of the Ethereum execution environment, always executing a specific piece of code when "poked" by a message or transaction, and having direct control over their own ether balance and their own key/value store to keep track of persistent variables.

i. Messages and Transactions

The term "transaction" is used in Ethereum to refer to the signed data package that stores a message to be sent from an externally owned account. Transactions contain:

- The recipient of the message
- A signature identifying the sender
- The amount of ether to transfer from the sender to the recipient
- An optional data field
- A `STARTGAS` value, representing the maximum number of computational steps the transaction execution is allowed to take
- A `GASPRICE` value, representing the fee the sender pays per computational step

The first three are standard fields expected in any cryptocurrency. The data field has no function by default, but the virtual machine has an opcode which a contract can use to access the data; as an example use case, if a contract is functioning as an on-blockchain domain registration service, then it may wish to interpret the data being passed to it as containing two "fields", the first field being a domain to register and the second field being the IP address to register it to. The contract would read these values from the message data and appropriately place them in storage.

The `STARTGAS` and `GASPRICE` fields are crucial for Ethereum's anti-denial of service model. In order to prevent accidental or hostile infinite loops or other computational wastage in code, each transaction is required to set a limit to how many computational steps of code execution it can use. The fundamental unit of computation is "gas"; usually, a computational step costs 1 gas, but some operations cost higher amounts of gas because they are more computationally expensive, or increase the amount of data that must be stored as part of the state. There is also a fee of 5 gas for every byte in the transaction data. The intent of the fee system is to require an attacker to pay proportionately for every resource that they consume, including computation, bandwidth and storage; hence, any transaction that leads to the network consuming a greater amount of any of these resources must have a gas fee roughly proportional to the increment.

i. Messages

Contracts have the ability to send "messages" to other contracts. Messages are virtual objects that are never serialized and exist only in the Ethereum execution environment. A message contains:

- The sender of the message (implicit)
- The recipient of the message
- The amount of ether to transfer alongside the message
- An optional data field
- A `STARTGAS` value

Essentially, a message is like a transaction, except it is produced by a contract and not an external actor. A message is produced when a contract currently executing code executes the `CALL` opcode, which produces and executes a message. Like a transaction, a

message leads to the recipient account running its code. Thus, contracts can have relationships with other contracts in exactly the same way that external actors can.

Note that the gas allowance assigned by a transaction or contract applies to the total gas consumed by that transaction and all sub-executions. For example, if an external actor A sends a transaction to B with 1000 gas, and B consumes 600 gas before sending a message to C, and the internal execution of C consumes 300 gas before returning, then B can spend another 100 gas before running out of gas.

i. Ethereum State Transition Function

![ethertransition.png](<https://raw.githubusercontent.com/ethereumbuilders/GitBook/master/en/vitalik-diagrams/ethertransition.png>)

The Ethereum state transition function, `APPLY(S,TX) -> S` can be defined as follows:

1. Check if the transaction is well-formed (ie. has the right number of values), the signature is valid, and the nonce matches the nonce in the sender's account. If not, return an error.
2. Calculate the transaction fee as `STARTGAS * GASPRICE`, and determine the sending address from the signature. Subtract the fee from the sender's account balance and increment the sender's nonce. If there is not enough balance to spend, return an error.
3. Initialize `GAS = STARTGAS`, and take off a certain quantity of gas per byte to pay for the bytes in the transaction.
4. Transfer the transaction value from the sender's account to the receiving account. If the receiving account does not yet exist, create it. If the receiving account is a contract, run the contract's code either to completion or until the execution runs out of gas.
5. If the value transfer failed because the sender did not have enough money, or the code execution ran out of gas, revert all state changes except the payment of the fees, and add the fees to the miner's account.
6. Otherwise, refund the fees for all remaining gas to the sender, and send the fees paid for gas consumed to the miner.

For example, suppose that the contract's code is:

```
if !self.storage[calldataload(0)]:  
    self.storage[calldataload(0)] = calldataload(32)
```

Note that in reality the contract code is written in the low-level EVM code; this example is written in Serpent, one of our high-level languages, for clarity, and can be compiled down to EVM code. Suppose that the contract's storage starts off empty, and a transaction is sent with 10 ether value, 2000 gas, 0.001 ether gasprice, and 64 bytes of data, with bytes 0-31 representing the number `2` and bytes 32-63 representing the string `CHARLIE`.^[fn.] (<https://github.com/ethereum/wiki/wiki/White-Paper#notes>) The process for the state transition function in this case is as follows:

1. Check that the transaction is valid and well formed.
2. Check that the transaction sender has at least $2000 * 0.001 = 2$ ether. If it is, then subtract 2 ether from the sender's account.
3. Initialize gas = 2000; assuming the transaction is 170 bytes long and the byte-fee is 5, subtract 850 so that there is 1150 gas left.
3. Subtract 10 more ether from the sender's account, and add it to the contract's account.
4. Run the code. In this case, this is simple: it checks if the contract's storage at index `2` is used, notices that it is not, and so it sets the storage at index `2` to the value `CHARLIE`. Suppose this takes 187 gas, so the remaining amount of gas is $1150 - 187 = 963$
5. Add $963 * 0.001 = 0.963$ ether back to the sender's account, and return the resulting state.

If there was no contract at the receiving end of the transaction, then the total transaction fee would simply be equal to the provided `GASPRICE` multiplied by the length of the transaction in bytes, and the data sent alongside the transaction would be irrelevant.

Note that messages work equivalently to transactions in terms of reverts: if a message execution runs out of gas, then that message's execution, and all other executions triggered by that execution, revert, but parent executions do not need to revert. This means that it is "safe" for a contract to call another contract, as if A calls B with G gas then A's execution is guaranteed to lose at most G gas. Finally, note that there is an opcode, `CREATE`, that creates a contract; its execution mechanics are generally similar to `CALL` ,

with the exception that the output of the execution determines the code of a newly created contract.

i. Code Execution

The code in Ethereum contracts is written in a low-level, stack-based bytecode language, referred to as "Ethereum virtual machine code" or "EVM code". The code consists of a series of bytes, where each byte represents an operation. In general, code execution is an infinite loop that consists of repeatedly carrying out the operation at the current program counter (which begins at zero) and then incrementing the program counter by one, until the end of the code is reached or an error or `STOP` or `RETURN` instruction is detected. The operations have access to three types of space in which to store data:

- The **stack**, a last-in-first-out container to which values can be pushed and popped
- **Memory**, an infinitely expandable byte array
- The contract's long-term **storage**, a key/value store. Unlike stack and memory, which reset after computation ends, storage persists for the long term.

The code can also access the value, sender and data of the incoming message, as well as block header data, and the code can also return a byte array of data as an output.

The formal execution model of EVM code is surprisingly simple. While the Ethereum virtual machine is running, its full computational state can be defined by the tuple `(block_state, transaction, message, code, memory, stack, pc, gas)`, where `block_state` is the global state containing all accounts and includes balances and storage. At the start of every round of execution, the current instruction is found by taking the `pc`-th byte of `code` (or 0 if `pc >= len(code)`), and each instruction has its own definition in terms of how it affects the tuple. For example, `ADD` pops two items off the stack and pushes their sum, reduces `gas` by 1 and increments `pc` by 1, and `SSTORE` pops the top two items off the stack and inserts the second item into the contract's storage at the index specified by the first item. Although there are many ways to optimize Ethereum virtual machine execution via just-in-time

compilation, a basic implementation of Ethereum can be done in a few hundred lines of code.

i. Blockchain and Mining

![apply_block_diagram.png](https://raw.githubusercontent.com/ethereumbuilders/GitBook/master/en/vitalik-diagrams/apply_block_diagram.png)

The Ethereum blockchain is in many ways similar to the Bitcoin blockchain, although it does have some differences. The main difference between Ethereum and Bitcoin with regard to the blockchain architecture is that, unlike Bitcoin(which only contains a copy of the transaction list), Ethereum blocks contain a copy of both the transaction list and the most recent state. Aside from that, two other values, the block number and the difficulty, are also stored in the block. The basic block validation algorithm in Ethereum is as follows:

1. Check if the previous block referenced exists and is valid.
2. Check that the timestamp of the block is greater than that of the referenced previous block and less than 15 minutes into the future
3. Check that the block number, difficulty, transaction root, uncle root and gas limit (various low-level Ethereum-specific concepts) are valid.
4. Check that the proof of work on the block is valid.
5. Let `S[0]` be the state at the end of the previous block.
6. Let `TX` be the block's transaction list, with `n` transactions. For all `i` in `0...n-1`, set `S[i+1] = APPLY(S[i], TX[i])`. If any application returns an error, or if the total gas consumed in the block up until this point exceeds the `GASLIMIT`, return an error.
7. Let `S_FINAL` be `S[n]`, but adding the block reward paid to the miner.
8. Check if the Merkle tree root of the state `S_FINAL` is equal to the final state root provided in the block header. If it is, the block is valid; otherwise, it is not valid.

The approach may seem highly inefficient at first glance, because it needs to store the entire state with each block, but in reality efficiency should be comparable to that of Bitcoin. The reason is that the state is stored in the tree structure, and after every block only a small part of the tree needs to be changed. Thus, in general,

between two adjacent blocks the vast majority of the tree should be the same, and therefore the data can be stored once and referenced twice using pointers (ie. hashes of subtrees). A special kind of tree known as a "Patricia tree" is used to accomplish this, including a modification to the Merkle tree concept that allows for nodes to be inserted and deleted, and not just changed, efficiently. Additionally, because all of the state information is part of the last block, there is no need to store the entire blockchain history - a strategy which, if it could be applied to Bitcoin, can be calculated to provide 5-20x savings in space.

A commonly asked question is "where" contract code is executed, in terms of physical hardware. This has a simple answer: the process of executing contract code is part of the definition of the state transition function, which is part of the block validation algorithm, so if a transaction is added into block `B` the code execution spawned by that transaction will be executed by all nodes, now and in the future, that download and validate block `B`.

1. Applications

In general, there are three types of applications on top of Ethereum. The first category is financial applications, providing users with more powerful ways of managing and entering into contracts using their money. This includes sub-currencies, financial derivatives, hedging contracts, savings wallets, wills, and ultimately even some classes of full-scale employment contracts. The second category is semi-financial applications, where money is involved but there is also a heavy non-monetary side to what is being done; a perfect example is self-enforcing bounties for solutions to computational problems. Finally, there are applications such as online voting and decentralized governance that are not financial at all.

i. Token Systems

On-blockchain token systems have many applications ranging from sub-currencies representing assets such as USD or gold to company stocks, individual tokens representing smart property, secure unforgeable coupons, and even token systems with no ties to conventional value at all, used as point systems for incentivization. Token systems are surprisingly easy to implement in

Ethereum. The key point to understand is that a currency, or token system, fundamentally is a database with one operation: subtract X units from A and give X units to B, with the provision that (1) A had at least X units before the transaction and (2) the transaction is approved by A. All that it takes to implement a token system is to implement this logic into a contract.

The basic code for implementing a token system in Serpent looks as follows:

```
def send(to, value):
    if self.storage[msg.sender] >= value:
        self.storage[msg.sender] =
self.storage[msg.sender] - value
        self.storage[to] = self.storage[to] + value
```

This is essentially a literal implementation of the "banking system" state transition function described further above in this document. A few extra lines of code need to be added to provide for the initial step of distributing the currency units in the first place and a few other edge cases, and ideally a function would be added to let other contracts query for the balance of an address. But that's all there is to it. Theoretically, Ethereum-based token systems acting as sub-currencies can potentially include another important feature that on-chain Bitcoin-based meta-currencies lack: the ability to pay transaction fees directly in that currency. The way this would be implemented is that the contract would maintain an ether balance with which it would refund ether used to pay fees to the sender, and it would refill this balance by collecting the internal currency units that it takes in fees and reselling them in a constant running auction. Users would thus need to "activate" their accounts with ether, but once the ether is there it would be reusable because the contract would refund it each time.

i. Financial derivatives and Stable-Value Currencies

Financial derivatives are the most common application of a "smart contract", and one of the simplest to implement in code. The main challenge in implementing financial contracts is that the majority of them require reference to an external price ticker; for example, a very desirable application is a smart contract that hedges against

the volatility of ether (or another cryptocurrency) with respect to the US dollar, but doing this requires the contract to know what the value of ETH/USD is. The simplest way to do this is through a "data feed" contract maintained by a specific party (eg. NASDAQ) designed so that that party has the ability to update the contract as needed, and providing an interface that allows other contracts to send a message to that contract and get back a response that provides the price.

Given that critical ingredient, the hedging contract would look as follows:

1. Wait for party A to input 1000 ether.
2. Wait for party B to input 1000 ether.
3. Record the USD value of 1000 ether, calculated by querying the data feed contract, in storage, say this is \$x.
4. After 30 days, allow A or B to "reactivate" the contract in order to send \$x worth of ether (calculated by querying the data feed contract again to get the new price) to A and the rest to B.

Such a contract would have significant potential in crypto-commerce. One of the main problems cited about cryptocurrency is the fact that it's volatile; although many users and merchants may want the security and convenience of dealing with cryptographic assets, they may not wish to face that prospect of losing 23% of the value of their funds in a single day. Up until now, the most commonly proposed solution has been issuer-backed assets; the idea is that an issuer creates a sub-currency in which they have the right to issue and revoke units, and provide one unit of the currency to anyone who provides them (offline) with one unit of a specified underlying asset (eg. gold, USD). The issuer then promises to provide one unit of the underlying asset to anyone who sends back one unit of the crypto-asset. This mechanism allows any non-cryptographic asset to be "uplifted" into a cryptographic asset, provided that the issuer can be trusted.

In practice, however, issuers are not always trustworthy, and in some cases the banking infrastructure is too weak, or too hostile, for such services to exist. Financial derivatives provide an alternative. Here, instead of a single issuer providing the funds to

back up an asset, a decentralized market of speculators, betting that the price of a cryptographic reference asset (eg. ETH) will go up, plays that role. Unlike issuers, speculators have no option to default on their side of the bargain because the hedging contract holds their funds in escrow. Note that this approach is not fully decentralized, because a trusted source is still needed to provide the price ticker, although arguably even still this is a massive improvement in terms of reducing infrastructure requirements (unlike being an issuer, issuing a price feed requires no licenses and can likely be categorized as free speech) and reducing the potential for fraud.

i. Identity and Reputation Systems

The earliest alternative cryptocurrency of all, [Namecoin](<http://namecoin.org/>), attempted to use a Bitcoin-like blockchain to provide a name registration system, where users can register their names in a public database alongside other data. The major cited use case is for a [DNS](http://en.wikipedia.org/wiki/Domain_Name_System) system, mapping domain names like "bitcoin.org" (or, in Namecoin's case, "bitcoin.bit") to an IP address. Other use cases include email authentication and potentially more advanced reputation systems. Here is the basic contract to provide a Namecoin-like name registration system on Ethereum:

```
def register(name, value):
    if !self.storage[name]:
        self.storage[name] = value
```

The contract is very simple; all it is a database inside the Ethereum network that can be added to, but not modified or removed from. Anyone can register a name with some value, and that registration then sticks forever. A more sophisticated name registration contract will also have a "function clause" allowing other contracts to query it, as well as a mechanism for the "owner" (ie. the first registerer) of a name to change the data or transfer ownership. One can even add reputation and web-of-trust functionality on top.

i. Decentralized File Storage

Over the past few years, there have emerged a number of popular online file storage startups, the most prominent being Dropbox,

seeking to allow users to upload a backup of their hard drive and have the service store the backup and allow the user to access it in exchange for a monthly fee. However, at this point the file storage market is at times relatively inefficient; a cursory look at various [existing](<http://online-storage-service-review.toptenreviews.com/>) shows that, particularly at the "uncanny valley" 20-200 GB level at which neither free quotas nor enterprise-level discounts kick in, monthly prices for mainstream file storage costs are such that you are paying for more than the cost of the entire hard drive in a single month. Ethereum contracts can allow for the development of a decentralized file storage ecosystem, where individual users can earn small quantities of money by renting out their own hard drives and unused space can be used to further drive down the costs of file storage.

The key underpinning piece of such a device would be what we have termed the "decentralized Dropbox contract". This contract works as follows. First, one splits the desired data up into blocks, encrypting each block for privacy, and builds a Merkle tree out of it. One then makes a contract with the rule that, every N blocks, the contract would pick a random index in the Merkle tree (using the previous block hash, accessible from contract code, as a source of randomness), and give X ether to the first entity to supply a transaction with a simplified payment verification-like proof of ownership of the block at that particular index in the tree. When a user wants to re-download their file, they can use a micropayment channel protocol (eg. pay 1 szabo per 32 kilobytes) to recover the file; the most fee-efficient approach is for the payer not to publish the transaction until the end, instead replacing the transaction with a slightly more lucrative one with the same nonce after every 32 kilobytes.

An important feature of the protocol is that, although it may seem like one is trusting many random nodes not to decide to forget the file, one can reduce that risk down to near-zero by splitting the file into many pieces via secret sharing, and watching the contracts to see each piece is still in some node's possession. If a contract is still paying out money, that provides a cryptographic proof that someone out there is still storing the file.

i. Decentralized Autonomous Organizations

The general concept of a "decentralized autonomous organization" is that of a virtual entity that has a certain set of members or shareholders which, perhaps with a 67% majority, have the right to spend the entity's funds and modify its code. The members would collectively decide on how the organization should allocate its funds. Methods for allocating a DAO's funds could range from bounties, salaries to even more exotic mechanisms such as an internal currency to reward work. This essentially replicates the legal trappings of a traditional company or nonprofit but using only cryptographic blockchain technology for enforcement. So far much of the talk around DAOs has been around the "capitalist" model of a "decentralized autonomous corporation" (DAC) with dividend-receiving shareholders and tradable shares; an alternative, perhaps described as a "decentralized autonomous community", would have all members have an equal share in the decision making and require 67% of existing members to agree to add or remove a member. The requirement that one person can only have one membership would then need to be enforced collectively by the group.

A general outline for how to code a DAO is as follows. The simplest design is simply a piece of self-modifying code that changes if two thirds of members agree on a change. Although code is theoretically immutable, one can easily get around this and have de-facto mutability by having chunks of the code in separate contracts, and having the address of which contracts to call stored in the modifiable storage. In a simple implementation of such a DAO contract, there would be three transaction types, distinguished by the data provided in the transaction:

- `'[0,i,K,V]` to register a proposal with index `i` to change the address at storage index `K` to value `V`
- `'[1,i]` to register a vote in favor of proposal `i`
- `'[2,i]` to finalize proposal `i` if enough votes have been made

The contract would then have clauses for each of these. It would maintain a record of all open storage changes, along with a list of who voted for them. It would also have a list of all members. When any storage change gets to two thirds of members voting for it, a

finalizing transaction could execute the change. A more sophisticated skeleton would also have built-in voting ability for features like sending a transaction, adding members and removing members, and may even provide for [Liquid](http://en.wikipedia.org/wiki/Delegative_democracy)-style vote delegation (ie. anyone can assign someone to vote for them, and assignment is transitive so if A assigns B and B assigns C then C determines A's vote). This design would allow the DAO to grow organically as a decentralized community, allowing people to eventually delegate the task of filtering out who is a member to specialists, although unlike in the "current system" specialists can easily pop in and out of existence over time as individual community members change their alignments.

An alternative model is for a decentralized corporation, where any account can have zero or more shares, and two thirds of the shares are required to make a decision. A complete skeleton would involve asset management functionality, the ability to make an offer to buy or sell shares, and the ability to accept offers (preferably with an order-matching mechanism inside the contract). Delegation would also exist Liquid Democracy-style, generalizing the concept of a "board of directors".

- i. Further Applications

- 1. Savings wallets**. Suppose that Alice wants to keep her funds safe, but is worried that she will lose or someone will hack her private key. She puts ether into a contract with Bob, a bank, as follows:

- Alice alone can withdraw a maximum of 1% of the funds per day.
- Bob alone can withdraw a maximum of 1% of the funds per day, but Alice has the ability to make a transaction with her key shutting off this ability.
- Alice and Bob together can withdraw anything.

Normally, 1% per day is enough for Alice, and if Alice wants to withdraw more she can contact Bob for help. If Alice's key gets hacked, she runs to Bob to move the funds to a new contract. If she loses her key, Bob will get the funds out eventually. If Bob turns out to be malicious, then she can turn off his ability to withdraw.

- 2. Crop insurance**. One can easily make a financial derivatives contract but using a data feed of the weather instead of any price index. If a farmer in Iowa purchases a derivative that pays out inversely based on the precipitation in Iowa, then if there is a drought, the farmer will automatically receive money and if there is enough rain the farmer will be happy because their crops would do well. This can be expanded to natural disaster insurance generally.
- 3. A decentralized data feed**. For financial contracts for difference, it may actually be possible to decentralize the data feed via a protocol called [SchellingCoin](<http://blog.ethereum.org/2014/03/28/schellingcoin-a-minimal-trust-universal-data-feed/>). SchellingCoin basically works as follows: N parties all put into the system the value of a given datum (eg. the ETH/USD price), the values are sorted, and everyone between the 25th and 75th percentile gets one token as a reward. Everyone has the incentive to provide the answer that everyone else will provide, and the only value that a large number of players can realistically agree on is the obvious default: the truth. This creates a decentralized protocol that can theoretically provide any number of values, including the ETH/USD price, the temperature in Berlin or even the result of a particular hard computation.
- 4. Smart multisignature escrow**. Bitcoin allows multisignature transaction contracts where, for example, three out of a given five keys can spend the funds. Ethereum allows for more granularity; for example, four out of five can spend everything, three out of five can spend up to 10% per day, and two out of five can spend up to 0.5% per day. Additionally, Ethereum multisig is asynchronous - two parties can register their signatures on the blockchain at different times and the last signature will automatically send the transaction.
- 5. Cloud computing**. The EVM technology can also be used to create a verifiable computing environment, allowing users to ask others to carry out computations

and then optionally ask for proofs that computations at certain randomly selected checkpoints were done correctly. This allows for the creation of a cloud computing market where any user can participate with their desktop, laptop or specialized server, and spot-checking together with security deposits can be used to ensure that the system is trustworthy (ie. nodes cannot profitably cheat). Although such a system may not be suitable for all tasks; tasks that require a high level of inter-process communication, for example, cannot easily be done on a large cloud of nodes. Other tasks, however, are much easier to parallelize; projects like SETI@home, folding@home and genetic algorithms can easily be implemented on top of such a platform.

- 6. Peer-to-peer gambling**. Any number of peer-to-peer gambling protocols, such as Frank Stajano and Richard Clayton's [Cyberdice](<http://www.cl.cam.ac.uk/~fms27/papers/2008-StajanoCla-cyberdice.pdf>), can be implemented on the Ethereum blockchain. The simplest gambling protocol is actually simply a contract for difference on the next block hash, and more advanced protocols can be built up from there, creating gambling services with near-zero fees that have no ability to cheat.
- 7. Prediction markets**. Provided an oracle or SchellingCoin, prediction markets are also easy to implement, and prediction markets together with SchellingCoin may prove to be the first mainstream application of [futarchy](<http://hanson.gmu.edu/futarchy.html>) as a governance protocol for decentralized organizations.
- 8. On-chain decentralized marketplaces**, using the identity and reputation system as a base.

1. Miscellanea And Concerns

i. Modified GHOST Implementation

The "Greedy Heaviest Observed Subtree" (GHOST) protocol is an innovation first introduced by Yonatan Sompolinsky and Aviv Zohar in [December](<https://eprint.iacr.org/2013/881.pdf>). The motivation behind GHOST is that blockchains with fast confirmation times

currently suffer from reduced security due to a high stale rate - because blocks take a certain time to propagate through the network, if miner A mines a block and then miner B happens to mine another block before miner A's block propagates to B, miner B's block will end up wasted and will not contribute to network security. Furthermore, there is a centralization issue: if miner A is a mining pool with 30% hashpower and B has 10% hashpower, A will have a risk of producing a stale block 70% of the time (since the other 30% of the time A produced the last block and so will get mining data immediately) whereas B will have a risk of producing a stale block 90% of the time. Thus, if the block interval is short enough for the stale rate to be high, A will be substantially more efficient simply by virtue of its size. With these two effects combined, blockchains which produce blocks quickly are very likely to lead to one mining pool having a large enough percentage of the network hashpower to have de facto control over the mining process.

As described by Sompolsky and Zohar, GHOST solves the first issue of network security loss by including stale blocks in the calculation of which chain is the "longest"; that is to say, not just the parent and further ancestors of a block, but also the stale descendants of the block's ancestor (in Ethereum jargon, "uncles") are added to the calculation of which block has the largest total proof of work backing it. To solve the second issue of centralization bias, we go beyond the protocol described by Sompolsky and Zohar, and also provide block rewards to stakes: a stale block receives 87.5% of its base reward, and the nephew that includes the stale block receives the remaining 12.5%. Transaction fees, however, are not awarded to uncles.

Ethereum implements a simplified version of GHOST which only goes down seven levels. Specifically, it is defined as follows:

- A block must specify a parent, and it must specify 0 or more uncles
- An uncle included in block `B` must have the following properties:

- * It must be a direct child of the `k`-th generation ancestor of `B`, where `2 <= k <= 7`.
- * It cannot be an ancestor of `B`
- * An uncle must be a valid block header, but does not need to be a previously verified or even valid block
- * An uncle must be different from all uncles included in previous blocks and all other uncles included in the same block (non-double-inclusion)
- For every uncle `U` in block `B`, the miner of `B` gets an additional 3.125% added to its coinbase reward and the miner of U gets 93.75% of a standard coinbase reward.

This limited version of GHOST, with uncles includable only up to 7 generations, was used for two reasons. First, unlimited GHOST would include too many complications into the calculation of which uncles for a given block are valid. Second, unlimited GHOST with compensation as used in Ethereum removes the incentive for a miner to mine on the main chain and not the chain of a public attacker.

i. Fees

Because every transaction published into the blockchain imposes on the network the cost of needing to download and verify it, there is a need for some regulatory mechanism, typically involving transaction fees, to prevent abuse. The default approach, used in Bitcoin, is to have purely voluntary fees, relying on miners to act as the gatekeepers and set dynamic minimums. This approach has been received very favorably in the Bitcoin community particularly because it is "market-based", allowing supply and demand between miners and transaction senders determine the price. The problem with this line of reasoning is, however, that transaction processing is not a market; although it is intuitively attractive to construe transaction processing as a service that the miner is offering to the sender, in reality every transaction that a miner includes will need to be processed by every node in the network, so the vast majority of the cost of transaction processing is borne by third parties and not the miner that is making the decision of whether or not to include it. Hence, tragedy-of-the-commons problems are very likely to occur. However, as it turns out this flaw in the market-based mechanism, when given a particular inaccurate simplifying assumption,

magically cancels itself out. The argument is as follows. Suppose that:

1. A transaction leads to `k` operations, offering the reward `kR` to any miner that includes it where `R` is set by the sender and `k` and `R` are (roughly) visible to the miner beforehand.
2. An operation has a processing cost of `C` to any node (ie. all nodes have equal efficiency)
3. There are `N` mining nodes, each with exactly equal processing power (ie. `1/N` of total)
4. No non-mining full nodes exist.

A miner would be willing to process a transaction if the expected reward is greater than the cost. Thus, the expected reward is `kR/N` since the miner has a `1/N` chance of processing the next block, and the processing cost for the miner is simply `kC`. Hence, miners will include transactions where `kR/N > kC`, or `R > NC`. Note that `R` is the per-operation fee provided by the sender, and is thus a lower bound on the benefit that the sender derives from the transaction, and `NC` is the cost to the entire network together of processing an operation. Hence, miners have the incentive to include only those transactions for which the total utilitarian benefit exceeds the cost.

However, there are several important deviations from those assumptions in reality:

1. The miner does pay a higher cost to process the transaction than the other verifying nodes, since the extra verification time delays block propagation and thus increases the chance the block will become a stale.
2. There do exist non-mining full nodes.
3. The mining power distribution may end up radically egalitarian in practice.
4. Speculators, political enemies and crazies whose utility function includes causing harm to the network do exist, and they can cleverly set up contracts where their cost is much lower than the cost paid by other verifying nodes.

(1) provides a tendency for the miner to include fewer transactions, and (2) increases `NC`; hence, these two effects at least partially cancel each other out.[\[How?\]](https://github.com/ethereum/wiki/issues/)(<https://github.com/ethereum/wiki/issues/>)

[447#issuecomment-316972260](#) (3) and (4) are the major issue; to solve them we simply institute a floating cap: no block can have more operations than `BLK_LIMIT_FACTOR` times the long-term exponential moving average. Specifically:

```
blk.oplimit = floor((blk.parent.oplimit * (EMAFATOR - 1) + floor(parent.opcount * BLK_LIMIT_FACTOR)) / EMA_FACTOR)
```

`BLK_LIMIT_FACTOR` and `EMAFATOR` are constants that will be set to 65536 and 1.5 for the time being, but will likely be changed after further analysis.

There is another factor disincentivizing large block sizes in Bitcoin: blocks that are large will take longer to propagate, and thus have a higher probability of becoming stale. In Ethereum, highly gas-consuming blocks can also take longer to propagate both because they are physically larger and because they take longer to process the transaction state transitions to validate. This delay disincentive is a significant consideration in Bitcoin, but less so in Ethereum because of the GHOST protocol; hence, relying on regulated block limits provides a more stable baseline.

i. Computation And Turing-Completeness

An important note is that the Ethereum virtual machine is Turing-complete; this means that EVM code can encode any computation that can be conceivably carried out, including infinite loops. EVM code allows looping in two ways. First, there is a `JUMP` instruction that allows the program to jump back to a previous spot in the code, and a `JUMPI` instruction to do conditional jumping, allowing for statements like `while x < 27: x = x * 2`. Second, contracts can call other contracts, potentially allowing for looping through recursion. This naturally leads to a problem: can malicious users essentially shut miners and full nodes down by forcing them to enter into an infinite loop? The issue arises because of a problem in computer science known as the halting problem: there is no way to tell, in the general case, whether or not a given program will ever halt.

As described in the state transition section, our solution works by requiring a transaction to set a maximum number of computational

steps that it is allowed to take, and if execution takes longer computation is reverted but fees are still paid. Messages work in the same way. To show the motivation behind our solution, consider the following examples:

- An attacker creates a contract which runs an infinite loop, and then sends a transaction activating that loop to the miner. The miner will process the transaction, running the infinite loop, and wait for it to run out of gas. Even though the execution runs out of gas and stops halfway through, the transaction is still valid and the miner still claims the fee from the attacker for each computational step.
- An attacker creates a very long infinite loop with the intent of forcing the miner to keep computing for such a long time that by the time computation finishes a few more blocks will have come out and it will not be possible for the miner to include the transaction to claim the fee. However, the attacker will be required to submit a value for `STARTGAS` limiting the number of computational steps that execution can take, so the miner will know ahead of time that the computation will take an excessively large number of steps.
- An attacker sees a contract with code of some form like `send(A,contract.storage[A]); contract.storage[A] = 0`, and sends a transaction with just enough gas to run the first step but not the second (ie. making a withdrawal but not letting the balance go down). The contract author does not need to worry about protecting against such attacks, because if execution stops halfway through the changes they get reverted.
- A financial contract works by taking the median of nine proprietary data feeds in order to minimize risk. An attacker takes over one of the data feeds, which is designed to be modifiable via the variable-address-call mechanism described in the section on DAOs, and converts it to run an infinite loop, thereby attempting to force any attempts to claim funds from the financial contract to run out of gas. However, the financial contract can set a gas limit on the message to prevent this problem.

The alternative to Turing-completeness is Turing-incompleteness, where `JUMP` and `JUMPI` do not exist and only one copy of each contract is allowed to exist in the call stack at any given time. With this system, the fee system described and the uncertainties around the effectiveness of our solution might not be necessary, as the cost of executing a contract would be bounded above by its size.

Additionally, Turing-incompleteness is not even that big a limitation; out of all the contract examples we have conceived internally, so far only one required a loop, and even that loop could be removed by making 26 repetitions of a one-line piece of code. Given the serious implications of Turing-completeness, and the limited benefit, why not simply have a Turing-incomplete language? In reality, however, Turing-incompleteness is far from a neat solution to the problem. To see why, consider the following contracts:

```
C0: call(C1); call(C1);
```

```
C1: call(C2); call(C2);
```

```
C2: call(C3); call(C3);
```

```
...
```

```
C49: call(C50); call(C50);
```

```
C50: (run one step of a program and record the change  
in storage)
```

Now, send a transaction to A. Thus, in 51 transactions, we have a contract that takes up 250 computational steps. Miners could try to detect such logic bombs ahead of time by maintaining a value alongside each contract specifying the maximum number of computational steps that it can take, and calculating this for contracts calling other contracts recursively, but that would require miners to forbid contracts that create other contracts (since the creation and execution of all 26 contracts above could easily be rolled into a single contract). Another problematic point is that the address field of a message is a variable, so in general it may not even be possible to tell which other contracts a given contract will call ahead of time. Hence, all in all, we have a surprising conclusion: Turing-completeness is surprisingly easy to manage, and the lack of Turing-completeness is equally surprisingly difficult to manage unless the exact same controls are in place - but in that case why not just let the protocol be Turing-complete?

i. Currency And Issuance

The Ethereum network includes its own built-in currency, ether, which serves the dual purpose of providing a primary liquidity layer to allow for efficient exchange between various types of digital assets and, more importantly, of providing a mechanism for paying transaction fees. For convenience and to avoid future argument (see the current mBTC/uBTC/satoshi debate in Bitcoin), the denominations will be pre-labelled:

- 1: wei
- 10₁₂: szabo
- 10₁₅: finney
- 10₁₈: ether

This should be taken as an expanded version of the concept of "dollars" and "cents" or "BTC" and "satoshi". In the near future, we expect "ether" to be used for ordinary transactions, "finney" for microtransactions and "szabo" and "wei" for technical discussions around fees and protocol implementation; the remaining denominations may become useful later and should not be included in clients at this point.

The issuance model will be as follows:

- Ether will be released in a currency sale at the price of 1000-2000 ether per BTC, a mechanism intended to fund the Ethereum organization and pay for development that has been used with success by other platforms such as Mastercoin and NXT. Earlier buyers will benefit from larger discounts. The BTC received from the sale will be used entirely to pay salaries and bounties to developers and invested into various for-profit and non-profit projects in the Ethereum and cryptocurrency ecosystem.
- 0.099x the total amount sold (60102216 ETH) will be allocated to the organization to compensate early contributors and pay ETH-denominated expenses before the genesis block.
- 0.099x the total amount sold will be maintained as a long-term reserve.
- 0.26x the total amount sold will be allocated to miners per year forever after that point.

I	Group I	At launch	I	After 1 year	I	After 5 years	I	-----	I		
-----	-----	-----	-----	-----	I	Currency units	I	1.198X	I	1.458X	
2.498X	I	Purchasers	I	83.5%	I	68.6%	I	40.0%	I	Reserve spent pre-sale	
I	8.26%	I	6.79%	I	3.96%	I	Reserve used post-sale	I	8.26%		
I	6.79%	I	3.96%	I	Miners	I	0%	I	17.8%	I	52.0%

- Long-Term Supply Growth Rate (percent)**

![SPV](<https://raw.githubusercontent.com/ethereumbuilders/GitBook/master/en/vitalik-diagrams/inflation.png>)

Despite the linear currency issuance, just like with Bitcoin over time the supply growth rate nevertheless tends to zero

The two main choices in the above model are (1) the existence and size of an endowment pool, and (2) the existence of a permanently growing linear supply, as opposed to a capped supply as in Bitcoin. The justification of the endowment pool is as follows. If the endowment pool did not exist, and the linear issuance reduced to 0.217x to provide the same inflation rate, then the total quantity of ether would be 16.5% less and so each unit would be 19.8% more valuable. Hence, in the equilibrium 19.8% more ether would be purchased in the sale, so each unit would once again be exactly as valuable as before. The organization would also then have 1.198x as much BTC, which can be considered to be split into two slices: the original BTC, and the additional 0.198x. Hence, this situation is exactly equivalent to the endowment, but with one important difference: the organization holds purely BTC, and so is not incentivized to support the value of the ether unit.

The permanent linear supply growth model reduces the risk of what some see as excessive wealth concentration in Bitcoin, and gives individuals living in present and future eras a fair chance to acquire currency units, while at the same time retaining a strong incentive to obtain and hold ether because the "supply growth rate" as a percentage still tends to zero over time. We also theorize that because coins are always lost over time due to carelessness, death, etc, and coin loss can be modeled as a percentage of the total supply per year, that the total currency supply in circulation will in fact eventually stabilize at a value equal to the annual issuance divided by the loss rate (eg. at a loss rate of 1%, once the supply

reaches 26X then 0.26X will be mined and 0.26X lost every year, creating an equilibrium).

Note that in the future, it is likely that Ethereum will switch to a proof-of-stake model for security, reducing the issuance requirement to somewhere between zero and 0.05X per year. In the event that the Ethereum organization loses funding or for any other reason disappears, we leave open a "social contract": anyone has the right to create a future candidate version of Ethereum, with the only condition being that the quantity of ether must be at most equal to $60102216 * (1.198 + 0.26 * n)$ where n is the number of years after the genesis block. Creators are free to crowd-sell or otherwise assign some or all of the difference between the PoS-driven supply expansion and the maximum allowable supply expansion to pay for development. Candidate upgrades that do not comply with the social contract may justifiably be forked into compliant versions.

i. Mining Centralization

The Bitcoin mining algorithm works by having miners compute SHA256 on slightly modified versions of the block header millions of times over and over again, until eventually one node comes up with a version whose hash is less than the target (currently around 2^{192}). However, this mining algorithm is vulnerable to two forms of centralization. First, the mining ecosystem has come to be dominated by ASICs (application-specific integrated circuits), computer chips designed for, and therefore thousands of times more efficient at, the specific task of Bitcoin mining. This means that Bitcoin mining is no longer a highly decentralized and egalitarian pursuit, requiring millions of dollars of capital to effectively participate in. Second, most Bitcoin miners do not actually perform block validation locally; instead, they rely on a centralized mining pool to provide the block headers. This problem is arguably worse: as of the time of this writing, the top three mining pools indirectly control roughly 50% of processing power in the Bitcoin network, although this is mitigated by the fact that miners can switch to other mining pools if a pool or coalition attempts a 51% attack. The current intent at Ethereum is to use a mining algorithm where miners are required to fetch random data from the state, compute

some randomly selected transactions from the last N blocks in the blockchain, and return the hash of the result. This has two important benefits. First, Ethereum contracts can include any kind of computation, so an Ethereum ASIC would essentially be an ASIC for general computation - ie. a better CPU. Second, mining requires access to the entire blockchain, forcing miners to store the entire blockchain and at least be capable of verifying every transaction. This removes the need for centralized mining pools; although mining pools can still serve the legitimate role of evening out the randomness of reward distribution, this function can be served equally well by peer-to-peer pools with no central control.

This model is untested, and there may be difficulties along the way in avoiding certain clever optimizations when using contract execution as a mining algorithm. However, one notably interesting feature of this algorithm is that it allows anyone to "poison the well", by introducing a large number of contracts into the blockchain specifically designed to stymie certain ASICs. The economic incentives exist for ASIC manufacturers to use such a trick to attack each other. Thus, the solution that we are developing is ultimately an adaptive economic human solution rather than purely a technical one.

i. Scalability

One common concern about Ethereum is the issue of scalability. Like Bitcoin, Ethereum suffers from the flaw that every transaction needs to be processed by every node in the network. With Bitcoin, the size of the current blockchain rests at about 15 GB, growing by about 1 MB per hour. If the Bitcoin network were to process Visa's 2000 transactions per second, it would grow by 1 MB per three seconds (1 GB per hour, 8 TB per year). Ethereum is likely to suffer a similar growth pattern, worsened by the fact that there will be many applications on top of the Ethereum blockchain instead of just a currency as is the case with Bitcoin, but ameliorated by the fact that Ethereum full nodes need to store just the state instead of the entire blockchain history.

The problem with such a large blockchain size is centralization risk. If the blockchain size increases to, say, 100 TB, then the likely

scenario would be that only a very small number of large businesses would run full nodes, with all regular users using light SPV nodes. In such a situation, there arises the potential concern that the full nodes could band together and all agree to cheat in some profitable fashion (eg. change the block reward, give themselves BTC). Light nodes would have no way of detecting this immediately. Of course, at least one honest full node would likely exist, and after a few hours information about the fraud would trickle out through channels like Reddit, but at that point it would be too late: it would be up to the ordinary users to organize an effort to blacklist the given blocks, a massive and likely infeasible coordination problem on a similar scale as that of pulling off a successful 51% attack. In the case of Bitcoin, this is currently a problem, but there exists a blockchain modification [suggested] (<https://web.archive.org/web/20140623061815/http://sourceforge.net/p/bitcoin/mailman/message/31709140/>) which will alleviate this issue.

In the near term, Ethereum will use two additional strategies to cope with this problem. First, because of the blockchain-based mining algorithms, at least every miner will be forced to be a full node, creating a lower bound on the number of full nodes. Second and more importantly, however, we will include an intermediate state tree root in the blockchain after processing each transaction. Even if block validation is centralized, as long as one honest verifying node exists, the centralization problem can be circumvented via a verification protocol. If a miner publishes an invalid block, that block must either be badly formatted, or the state `S[n]` is incorrect. Since `S[0]` is known to be correct, there must be some first state `S[i]` that is incorrect where `S[i-1]` is correct. The verifying node would provide the index `i`, along with a "proof of invalidity" consisting of the subset of Patricia tree nodes needing to process `APPLY(S[i-1], TX[i]) -> S[i]`. Nodes would be able to use those Patricia nodes to run that part of the computation, and see that the `S[i]` generated does not match the `S[i]` provided.

Another, more sophisticated, attack would involve the malicious miners publishing incomplete blocks, so the full information does

not even exist to determine whether or not blocks are valid. The solution to this is a challenge-response protocol: verification nodes issue "challenges" in the form of target transaction indices, and upon receiving a node a light node treats the block as untrusted until another node, whether the miner or another verifier, provides a subset of Patricia nodes as a proof of validity.

1. Conclusion

The Ethereum protocol was originally conceived as an upgraded version of a cryptocurrency, providing advanced features such as on-blockchain escrow, withdrawal limits, financial contracts, gambling markets and the like via a highly generalized programming language. The Ethereum protocol would not "support" any of the applications directly, but the existence of a Turing-complete programming language means that arbitrary contracts can theoretically be created for any transaction type or application. What is more interesting about Ethereum, however, is that the Ethereum protocol moves far beyond just currency. Protocols around decentralized file storage, decentralized computation and decentralized prediction markets, among dozens of other such concepts, have the potential to substantially increase the efficiency of the computational industry, and provide a massive boost to other peer-to-peer protocols by adding for the first time an economic layer. Finally, there is also a substantial array of applications that have nothing to do with money at all.

The concept of an arbitrary state transition function as implemented by the Ethereum protocol provides for a platform with unique potential; rather than being a closed-ended, single-purpose protocol intended for a specific array of applications in data storage, gambling or finance, Ethereum is open-ended by design, and we believe that it is extremely well-suited to serving as a foundational layer for a very large number of both financial and non-financial protocols in the years to come.

1. Notes and Further Reading

i. Notes

1. A sophisticated reader may notice that in fact a Bitcoin address is the hash of the elliptic curve public key, and not the public key itself.

However, it is in fact perfectly legitimate cryptographic terminology to refer to the pubkey hash as a public key itself. This is because Bitcoin's cryptography can be considered to be a custom digital signature algorithm, where the public key consists of the hash of the ECC pubkey, the signature consists of the ECC pubkey concatenated with the ECC signature, and the verification algorithm involves checking the ECC pubkey in the signature against the ECC pubkey hash provided as a public key and then verifying the ECC signature against the ECC pubkey.

2. Technically, the median of the 11 previous blocks.

3. The Ethereum protocol should be as simple as practical, but it may be necessary to have quite a high level of complexity, for instance to scale, to internalize costs of storage, bandwidth and I/O, for security, privacy, transparency, etc. Where complexity is necessary, documentation should be as clear, concise and up-to-date as possible, so that someone completely unschooled in Ethereum can learn it and become an expert.

4. See the [Yellow](<https://ethereum.github.io/yellowpaper/paper.pdf>) for the Ethereum Virtual Machine (which is useful as a specification and as a reference for building an Ethereum client from scratch), while also there are many topics in the [Ethereum](<https://github.com/ethereum/wiki/wiki>), such as sharding development, core development, dapp development, research, Casper R&D, and networking protocols. For research and possible future implementation there is [ethresear.ch](<https://ethresear.ch>).

5. Another way of expressing this is abstraction. The [latest](<https://ethresear.ch/t/sharding-phase-1-spec/1407/67>) is planning to abstract execution, allowing execution engines to not necessarily have to follow one canonical specification, but for instance it could be tailored for a specific application, as well as a shard. (This heterogeneity of execution engines is not explicitly stated in the roadmap. There is also heterogeneous sharding, which Vlad Zamfir conceptualized.)

6. Internally, 2 and "CHARLIE" are both numbers, with the latter being in big-endian base 256 representation. Numbers can be at least 0 and at most $2^{256}-1$.

i. Further Reading

1. Intrinsic value: <http://bitcoinmagazine.com/8640/an-exploration-of-intrinsic-value-what-it-is-why-bitcoin-doesnt-have-it-and-why-bitcoin-does-have-it/>
2. Smart property: <https://en.bitcoin.it/wiki/>

[Smart Property](#) 3. Smart contracts: <https://en.bitcoin.it/wiki/Contracts> 4. B-money: <http://www.weidai.com/bmoney.txt>

5. Reusable proofs of work: <http://www.finney.org/~hal/rpow/> 6. Secure property titles with owner authority: <http://szabo.best.vwh.net/securetitle.html> 7. Bitcoin whitepaper: <http://bitcoin.org/bitcoin.pdf>

8. Namecoin: <https://namecoin.org/> 9. Zooko's triangle: http://en.wikipedia.org/wiki/Zooko%27s_triangle 10. Colored coins

whitepaper: https://docs.google.com/a/buterin.com/document/d/1AnkP_cVZTCMLIzw4DvsW6M8Q2JC0lIzrTLuoWu2z1BE/edit 11.

Mastercoin whitepaper: <https://github.com/mastercoin-MSC/spec> 12. Decentralized autonomous corporations, Bitcoin Magazine: <http://bitcoinmagazine.com/7050/bootstrapping-a-decentralized-autonomous-corporation-part-i/>

13. Simplified payment verification: <https://en.bitcoin.it/wiki/Scalability#Simplifiedpaymentverification> 14. Merkle trees: http://en.wikipedia.org/wiki/Merkle_tree 15. Patricia trees: http://en.wikipedia.org/wiki/Patricia_tree 16. GHOST: <https://eprint.iacr.org/2013/881.pdf>

17. StorJ and Autonomous Agents, Jeff Garzik: <http://garzikrants.blogspot.ca/2013/01/storj-and-bitcoin-autonomous-agents.html> 18. Mike Hearn on Smart Property at Turing

Festival: <http://www.youtube.com/watch?v=Pu4PAMFPo5Y> 19. Ethereum RLP: <https://github.com/ethereum/wiki/wiki/%5BEnglish%5D-RLP>

20. Ethereum Merkle Patricia trees: <https://github.com/ethereum/wiki/wiki/%5BEnglish%5D-Patricia-Tree> 21.

Peter Todd on Merkle sum trees: <http://sourceforge.net/p/bitcoin/mailman/message/31709140/>

For history of the white paper, see <https://github.com/ethereum/wiki/blob/old-before-deleting-all-files-go-to-wiki-wiki-instead/old-whitepaper-for-historical-reference.md#historical-sources-of-the-white-paper>

The Tangle

Serguei Popov*

April 30, 2018. Version 1.4.3

Abstract

In this paper we analyze the mathematical foundations of IOTA, a cryptocurrency for the Internet-of-Things (IoT) industry. The main feature of this novel cryptocurrency is the *tangle*, a directed acyclic graph (DAG) for storing transactions. The tangle naturally succeeds the blockchain as its next evolutionary step, and offers features that are required to establish a machine-to-machine micropayment system.

An essential contribution of this paper is a family of Markov Chain Monte Carlo (MCMC) algorithms. These algorithms select attachment sites on the tangle for a transaction that has just arrived.

1 Introduction and description of the system

The rise and success of Bitcoin during the last six years proved that blockchain technology has real-world value. However, this technology also has a number of drawbacks that prevent it from being used as a generic platform for cryptocurrencies across the globe. One notable drawback is the concept of a transaction fee for transactions of any value. The importance of micropayments will increase in the rapidly developing IoT industry, and paying a fee that is *larger* than the amount of value being transferred is not logical. Furthermore, it is not easy to get rid of fees in the blockchain infrastructure since they serve as an incentive for the creators of blocks. This leads to another issue with existing cryptocurrency technology, namely the heterogeneous nature of the system. There are two distinct types of participants in the system, those who issue transactions, and those who approve transactions. The design of this system creates unavoidable discrimination of some participants, which in turn creates

*a.k.a. `mthcl`; author's contact information: `serguei.popov@iota.org`

conflicts that make all elements spend resources on conflict resolution. The aforementioned issues justify a search for solutions essentially different from blockchain technology, the basis for Bitcoin and many other cryptocurrencies.

In this paper we discuss an innovative approach that does not incorporate blockchain technology. This approach is currently being implemented as a cryptocurrency called *iota* [1], which was designed specifically for the IoT industry. The purpose of this paper is to focus on general features of the tangle, and to discuss problems that arise when one attempts to get rid of the blockchain and maintain a distributed ledger. The concrete implementation of the iota protocol is not discussed.

In general, a tangle-based cryptocurrency works in the following way. Instead of the global blockchain, there is a DAG that we call the *tangle*. The transactions issued by nodes constitute the site set of the tangle graph, which is the ledger for storing transactions. The edge set of the tangle is obtained in the following way: when a new transaction arrives, it must *approve* two¹ previous transactions. These approvals are represented by directed edges, as shown in Figure 1². If there is not a directed edge between transaction *A* and transaction *B*, but there is a directed path of length at least two from *A* to *B*, we say that *A* *indirectly approves* *B*. There is also the “genesis” transaction, which is approved either directly or indirectly by all other transactions (Figure 2). The genesis is described in the following way. In the beginning of the tangle, there was an address with a balance that contained all of the tokens. The genesis transaction sent these tokens to several other “founder” addresses. Let us stress that all of the tokens were created in the genesis transaction. No tokens will be created in the future, and there will be no mining in the sense that miners receive monetary rewards “out of thin air”.

A quick note on terminology: *sites* are transactions represented on the tangle graph. The network is composed of *nodes*; that is, nodes are entities that issue and validate transactions.

The main idea of the tangle is the following: to issue a transaction, users must work to approve other transactions. Therefore, users who issue a transaction are contributing to the network’s security. It is assumed that the nodes check if the approved transactions are not conflicting. If a node finds that a transaction is in conflict with the tangle history, the node will not approve the conflicting transaction in either a direct or indirect manner³.

¹This is the simplest approach. One may also study similar systems where transactions must approve k other transactions for a general $k \geq 2$, or have an entirely different set of rules.

²Time always increases from left to right in each figure.

³If a node issues a new transaction that approves conflicting transactions, then it risks that other nodes will not approve its new transaction, which will fall into oblivion.

As a transaction receives additional approvals, it is accepted by the system with a higher level of confidence. In other words, it will be difficult to make the system accept a double-spending transaction. It is important to observe that we do not *impose* any rules for choosing which transactions a node will approve. Instead, we argue that if a large number of nodes follow some “reference” rule, then for any fixed node it is better to stick to a rule of the same kind⁴. This seems to be a reasonable assumption, especially in the context of IoT, where nodes are specialized chips with pre-installed firmware.

In order to issue a transaction, a node does the following:

- The node chooses two other transactions to approve according to an algorithm. In general, these two transactions may coincide.
- The node checks if the two transactions are not conflicting, and does not approve conflicting transactions.
- For a node to issue a valid transaction, the node must solve a cryptographic puzzle similar to those in the Bitcoin blockchain. This is achieved by finding a nonce such that the hash of that nonce concatenated with some data from the approved transaction has a particular form. In the case of the Bitcoin protocol, the hash must have at least a predefined number of leading zeros.

It is important to observe that the iota network is asynchronous. In general, nodes do not necessarily see the same set of transactions. It should also be noted that the tangle may contain conflicting transactions. The nodes do not have to achieve consensus on which valid⁵ transactions have the right to be in the ledger, meaning all of them can be in the tangle. However, in the case where there are conflicting transactions, the nodes need to decide which transactions will become orphaned⁶. The main rule that the nodes use for deciding between two conflicting transactions is the following: a node runs the tip selection algorithm⁷ (cf. Section 4.1) many times, and sees which of the two transactions is more likely to be indirectly approved by the selected tip. For example, if a transaction was selected 97 times during 100 runs of the tip selection algorithm, we say that it is confirmed with 97% confidence.

Let us also comment on the following question (cf. [4]): what motivates the nodes to propagate transactions? Every node calculates some statistics, one of which is

⁴We comment more on this at the end of Section 4.1

⁵Transactions that are issued according to the protocol.

⁶Orphaned transactions are not indirectly approved by incoming transactions anymore

⁷As mentioned above, there is a good reason to assume that other nodes would follow the same algorithm for tip selection.

how many new transactions are received from a neighbor. If one particular node is “too lazy”, it will be dropped by its neighbors. Therefore, even if a node does not issue transactions, and hence has no direct incentive to share new transactions that approve its own transaction, it still has incentive to participate.

After introducing some notation in Section 2, we discuss algorithms for choosing the two transactions to approve, the rules for measuring the transaction’s overall approval (Section 3, especially Section 3.1), and possible attack scenarios (Section 4). Also, in the unlikely event that the reader is scared by the formulas, they can jump directly to the “conclusions” at the end of each section.

It should be noted that the idea of using DAGs in the cryptocurrency space has been around for some time, see [3, 6, 7, 9, 12]. Specifically, [7] introduces the GHOST protocol, which proposes a modification of the Bitcoin protocol by making the main ledger a tree instead of a blockchain. It is shown that such a modification reduces confirmation times and improves the overall security of the network. In [9] the authors consider a DAG-based cryptocurrency model. Their model is different than our model for the following reasons: the sites of their DAG are blocks instead of individual transactions; the miners in their system compete for transaction fees; and new tokens may be created by block miners. Also, observe that a solution somewhat similar to ours was proposed in [6], although it does not discuss any particular tip approval strategies. After the first version of this paper was published, several other works that aim to create a DAG-based distributed ledger have appeared, e.g. [8]. We also reference another approach [2, 10] that aims to make Bitcoin micropayments possible by establishing peer-to-peer payment channels.

2 Weights and more

In this section we define the weight of a transaction, and related concepts. The weight of a transaction is proportional to the amount of work that the issuing node invested into it. In the current implementation of iota, the weight may only assume values 3^n , where n is a positive integer that belongs to some nonempty interval of acceptable values⁸. In fact, it is irrelevant to know how the weight was obtained in practice. It is only important that every transaction has a positive integer, its weight, attached to it. In general, the idea is that a transaction with a larger weight is more “important” than a transaction with a smaller weight. To avoid spamming and other attack styles, it is assumed that no entity can generate an abundance of transactions with “acceptable” weights in a short period of time.

⁸This interval should also be finite — see the “large weight attack” in Section 4.

One of the notions we need is the *cumulative weight* of a transaction: it is defined as the own weight of a particular transaction plus the sum of own weights of all transactions that directly or indirectly approve this transaction. The algorithm for cumulative weight calculation is illustrated in Figure 1. The boxes represent transactions, the small number in the SE corner of each box denotes own weight, and the bold number denotes the cumulative weight. For example, transaction F is directly or indirectly approved by transactions A, B, C, E . The cumulative weight of F is $9 = 3 + 1 + 3 + 1 + 1$, which is the sum of the own weight of F and the own weights of A, B, C, E .

Let us define “tips” as unapproved transactions in the tangle graph. In the top tangle snapshot of Figure 1, the only tips are A and C . When the new transaction X arrives and approves A and C in the bottom tangle snapshot, X becomes the only tip. The cumulative weight of all other transactions increases by 3, the own weight of X .

We need to introduce two additional variables for the discussion of approval algorithms. First, for a transaction site on the tangle, we introduce its

- *height*: the length of the longest oriented path to the genesis;
- *depth*: the length of the longest reverse-oriented path to some tip.

For example, G has height 1 and depth 4 in Figure 2 because of the reverse path F, D, B, A , while D has height 3 and depth 2. Also, let us introduce the notion of the *score*. By definition, the score of a transaction is the sum of own weights of all transactions approved by this transaction plus the own weight of the transaction itself. In Figure 2, the only tips are A and C . Transaction A directly or indirectly approves transactions B, D, F, G , so the score of A is $1+3+1+3+1 = 9$. Analogously, the score of C is $1+1+1+3+1 = 7$.

In order to understand the arguments presented in this paper, one may safely assume that all transactions have an own weight equal to 1. *From now on, we stick to this assumption.* Under this assumption, the cumulative weight of transaction X becomes 1 plus the number of transactions that directly or indirectly approve X , and the score becomes 1 plus the number of transactions that are directly or indirectly approved by X .

Let us note that, among those defined in this section, the cumulative weight is (by far!) the most important metric, although height, depth, and score will briefly enter some discussions as well.

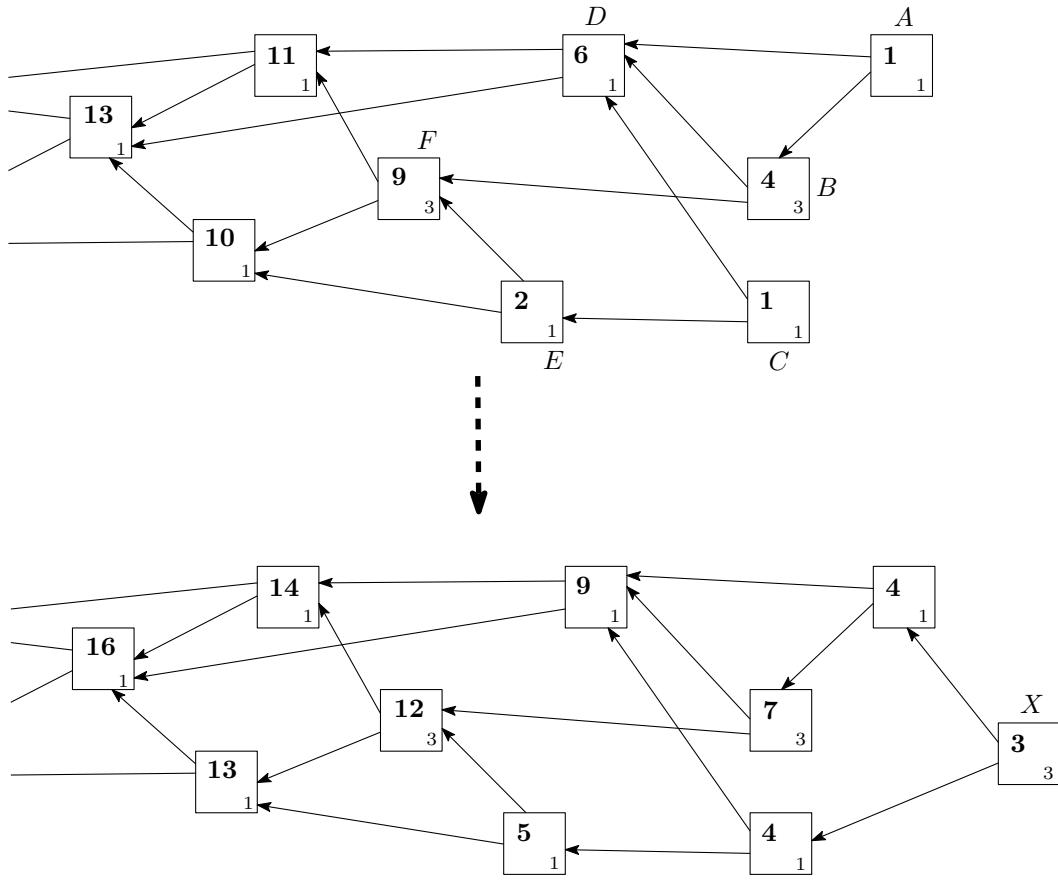


Figure 1: DAG with weight assignments before and after a newly issued transaction, X . The boxes represent transactions, the small number in the SE corner of each box denotes own weight, and the bold number denotes the cumulative weight.

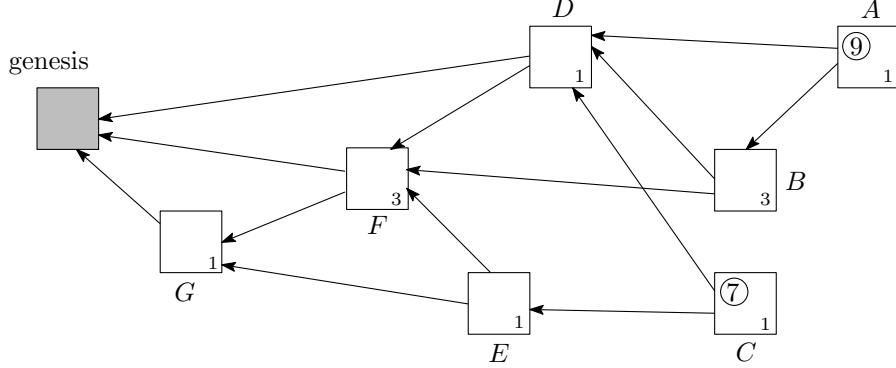


Figure 2: DAG with own weights assigned to each site, and scores calculated for sites A and C .

3 Stability of the system, and cutsets

Let $L(t)$ be the total number of tips in the system at time t . One expects that the stochastic process $L(t)$ remains *stable*⁹. More precisely, one expects the process to be *positive recurrent*, see Sections 4.4 and 6.5 of [11] for formal definitions. In particular, positive recurrence implies that the limit of $\mathbb{P}[L(t) = k]$ as $t \rightarrow \infty$ should exist and be positive for all $k \geq 1$. Intuitively, we expect that $L(t)$ should fluctuate around a constant value, and not escape to infinity. If $L(t)$ were to escape to infinity, many unapproved transactions would be left behind.

To analyze the stability properties of $L(t)$, we need to make some assumptions. One assumption is that transactions are issued by a large number of roughly independent entities, so the process of incoming transactions can be modeled by a Poisson point process (cf. e.g. Section 5.3 of [11]). Let λ be the rate of that Poisson process. For simplicity, let us assume that this rate remains constant in time. Assume that all devices have approximately the same computing power, and let h be the average time a device needs to perform calculations that are required to issue a transaction. Then, let us *assume* that all nodes behave in the following way: to issue a transaction, a node chooses two tips at random and approves them. It should be observed that, in general, it is *not* a good idea for the “honest nodes” to adopt this strategy because it has a number of practical disadvantages. In particular, it does not offer enough protection against “lazy” or malicious nodes (see Section 4.1 below). On the other hand, we still consider this model since it is simple to analyze, and may provide insight into the system’s behavior for more complicated tip selection strategies.

⁹Under an additional assumption that the process is time-homogeneous.

Next, we make a further simplifying assumption that any node, at the moment when it issues a transaction, observes not the actual state of the tangle, but the one exactly h time units ago. This means, in particular, that a transaction attached to the tangle at time t only becomes visible to the network at time $t+h$. We also assume that the number of tips remains roughly stationary in time, and is concentrated around a number $L_0 > 0$. In the following, we will calculate L_0 as a function of λ and h .

Observe that, at a given time t we have roughly λh “hidden tips” (which were attached in the time interval $[t-h, t)$ and so are not yet visible to the network); also, assume that typically there are r “revealed tips” (which were attached before time $t-h$ and remain tips at time t), so $L_0 = r + \lambda h$. By stationarity, we may then assume that at time t there are also around λh sites that were tips at time $t-h$, but are not tips anymore. Now, think about a new transaction that comes at this moment; then, a transaction it chooses to approve is a tip with probability $r/(r + \lambda h)$ (since there are around r tips known to the node that issued the transaction, and there are also around λh transactions which are not tips anymore, although that node thinks they are), so the mean number of chosen tips is $2r/(r + \lambda h)$. The key observation is now that, in the stationary regime, this mean number of chosen tips should be equal to 1, since, in average, a newcomer transaction should not change the number of tips. Solving the equation $2r/(r + \lambda h) = 1$ with respect to r , we obtain $r = \lambda h$, and so

$$L_0 = 2\lambda h. \quad (1)$$

We also note that, if the rule is that a new transaction references k transactions instead of 2, then a similar calculation gives

$$L_0^{(k)} = \frac{k\lambda h}{k-1}. \quad (2)$$

This is, of course, consistent with the fact that $L_0^{(k)}$ should tend to λh as $k \rightarrow \infty$ (basically, the only tips would be those still unknown to the network).

Also (we return to the case of two transactions to approve) the expected time for a transaction to be approved for the first time is approximately $h + L_0/2\lambda = 2h$. This is because, by our assumption, during the first h units of time a transaction cannot be approved, and after that the Poisson flow of approvals to it has rate approximately $2\lambda/L_0$. (Recall Proposition 5.3 of [11], which says that if we independently classify each event of a Poisson process according to a list of possible subtypes, then the processes of events of each subtype are independent Poisson processes.)

Observe that¹⁰ at any fixed time t the set of transactions that were tips at some

¹⁰At least in the case where the nodes *try* to approve tips.

moment $s \in [t, t + h(L_0, N)]$ typically constitutes a *cutset*. Any path from a transaction issued at time $t' > t$ to the genesis must pass through this set. It is important that the size of a new cutset in the tangle occasionally becomes small. One may then use the small cutsets as checkpoints for possible DAG pruning and other tasks.

It is important to observe that the above “purely random” approval strategy is not very good in practice because it does not encourage approving tips. A “lazy” user could always approve a fixed pair of very old transactions, therefore not contributing to the approval of more recent transactions, without being punished for such behavior¹¹. Also, a malicious entity can artificially inflate the number of tips by issuing many transactions that approve a fixed pair of transactions. This would make it possible for future transactions to select these tips with very high probability, effectively abandoning the tips belonging to “honest” nodes. To avoid issues of this sort, one has to adopt a strategy that is biased towards the “better” tips. One example of such a strategy is presented in Section 4.1 below.

Before starting the discussion about the expected time for a transaction to receive its first approval, note that we can distinguish two regimes (Figure 3).

- Low load: the typical number of tips is small, and frequently becomes 1. This may happen when the flow of transactions is so small that it is not probable that several different transactions approve the same tip. Also, if the network latency is very low and devices compute fast, it is unlikely that many tips would appear. This even holds true in the case when the flow of transactions is reasonably large. Moreover, we have to assume that there are no attackers that try to artificially inflate the number of tips.
- High load: the typical number of tips is large. This may happen when the flow of transactions is large, and computational delays together with network latency make it likely that several different transactions approve the same tip.

This division is rather informal, and there is no clear borderline between the two regimes. Nevertheless, we find that it may be instructive to consider these two different extremes.

The situation in the low load regime is relatively simple. The first approval happens on an average timescale of order λ^{-1} since one of the first few incoming transactions will approve a given tip.

Let us now consider the high load regime, the case where L_0 is large. As mentioned above, one may assume that the Poisson flows of approvals to different tips are

¹¹We remind the reader that we do not try to *enforce* any particular tip selection strategy. An attacker can choose tips in any way they find convenient.

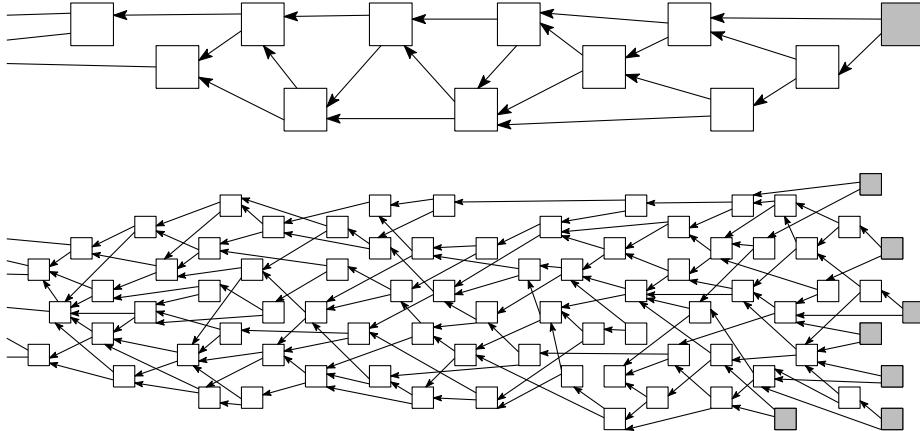


Figure 3: Low load (top) and high load (bottom) regimes of incoming transaction flow. White squares represent verified sites, while gray squares represent tips.

independent and have an approximate rate of $2\lambda/L_0$. Therefore, the expected time for a transaction to receive its first approval is around $L_0/(2\lambda) = h$ (recall (1)).

However, it is worth noting that for more elaborate approval strategies¹², it may not be a good idea to passively wait a long time until a transaction is approved by the others. This is due to the fact that “better” tips will keep appearing and will be preferred for approval. Rather, in the case when a transaction is waiting for approval over a time interval much larger than $L_0/2\lambda$, a good strategy would be to promote this latent transaction with an additional empty transaction¹³. In other words, a node can issue an empty transaction that approves its previous transaction together with one of the “better” tips to increase the probability that the empty transaction receives approval.

It turns out that the approval strategies based on heights and scores may be vulnerable to a specific type of attacks, see Section 4.1. We will discuss more elaborate strategies¹⁴ to defend against such attacks in that section. In the meantime, it is still

¹²That favor “better” quality tips in future implementations of iota.

¹³An empty transaction is a transaction that does not involve any token transfer, but still has to approve two other transactions. It should be noted that generating an empty transaction contributes to the network’s security.

¹⁴In fact, the author’s feeling is that the tip approval strategy is *the* most important ingredient for constructing a tangle-based cryptocurrency. It is there that many attack vectors are hiding. Also, since there is usually no way to *enforce* a particular tip approval strategy, it must be such that the nodes would voluntarily choose to follow it knowing that at least a good proportion of other nodes does so.

worth considering the simple tip selection strategy where an incoming transaction approves two random tips. This strategy is the easiest to analyze, and therefore may provide some insight into the qualitative and quantitative behavior of the tangle.

Conclusions:

1. We distinguish between two regimes, low load and high load (Figure 3).
2. There are only a few tips in the low load regime. A tip gets approved for the first time in $\Theta(\lambda^{-1})$ time units, where λ is the rate of the incoming flow of transactions.
3. In the high load regime the typical number of tips depends on the tip approval strategy employed by the new transaction.
4. If a transaction uses the strategy of approving two random tips, the typical number of tips is given by (1). It can be shown that this strategy is optimal with respect to the typical number of tips. However, it is not practical to adopt this strategy because it does not encourage approving tips.
5. More elaborate strategies are needed to handle attacks and other network issues. A family of such strategies is discussed in Section 4.1.
6. The typical time for a tip to be approved is $\Theta(h)$ in the high load regime, where h is the average computation/propagation time for a node. However, if the first approval does not occur in the above time interval, it is a good idea for the issuer and/or receiver to promote that transaction with an additional empty transaction.

3.1 How fast does the cumulative weight typically grow?

Assume that the network is in the low load regime. After a transaction gets approved several times, its cumulative weight will grow with speed λ because all new transactions will indirectly reference this transaction¹⁵.

In the case where the network is in the high load regime, an old transaction with a large cumulative weight will experience weight growth with speed λ because essentially all new transactions will indirectly reference it. Moreover, when the transaction

¹⁵Recall that we assumed that the own weights of all transactions are equal to 1, so the cumulative weight is just the number of transactions that directly or indirectly reference a transaction plus 1.

is first added to the tangle it may have to wait for some time to be approved. In this time interval, the transaction’s cumulative weight behaves in a random fashion. To characterize the speed with which the cumulative weight grows after the transaction receives several approvals, let us define $H(t)$ as the expected cumulative weight at time t (for simplicity, we start counting time at the moment when our transaction was revealed to the network, i.e., h time units after it was created) and $K(t)$ as the expected number of tips that approve the transaction at time t . Let us also abbreviate $h := h(L_0, N)$. We make a simplifying assumption that the number of tips remains roughly constant at a value of L_0 over time. We work with the “approve two random tips” strategy in this section. It is expected that the qualitative behavior will be roughly the same for other reasonable strategies.

Recall that a transaction entering the network at time t typically chooses two tips to approve based on the state of the system at time $t - h$ because the node must do some calculations and verifications before actually issuing the transaction. It is not difficult to see that (assuming, though, that $K(\cdot)$ is the *actual* number of tips, not just expected number) the probability of the transaction approving at least one of “our” tips in the tangle is $1 - (1 - \frac{K(t-h)}{L_0})^2 = \frac{K(t-h)}{L_0}(2 - \frac{K(t-h)}{L_0})$ ¹⁶. Analogous to Example 6.4 of [11], we can write for small $\delta > 0$

$$H(t + \delta) = H(t) + \lambda\delta \frac{K(t-h)}{L_0} \left(2 - \frac{K(t-h)}{L_0}\right) + o(\delta),$$

and thus deduce the following differential equation

$$\frac{dH(t)}{dt} = \lambda \frac{K(t-h)}{L_0} \left(2 - \frac{K(t-h)}{L_0}\right). \quad (3)$$

In order to be able to use (3), we need to first calculate $K(t)$. This is not a trivial task since a tip at time $t - h$ may not be a tip at time t , and the overall number of tips approving the original transaction increases by 1 in the case where an incoming transaction approves such a tip. The crucial observation is that the probability that a tip at time $t - h$ remains a tip at time t is approximately 1/2. (To verify this, recall the discussion from Section 3: the typical number of tips is $2\lambda h$, and during the interval of length h new λh tips will substitute a half of old ones.) Therefore, at time t approximately one half $K(t-h)$ tips remain in the unconfirmed tip state, while the other half will have received at least one approval. Let A denote the set of $K(t-h)/2$ tips at time $t - h$ that are still tips at time t , and let B denote the

¹⁶The expression on the left-hand side is 1 minus the probability that the two approved tips are not ours.

remaining set of $K(t-h)/2$ tips that were already approved by time t . Let p_1 be the probability that a new transaction approves at least 1 transaction from B and does not approve any transactions from A . Furthermore, let p_2 be the probability that both approved transactions belong to A . In other words, p_1 and p_2 are the probabilities that the current number of “our” tips increases or decreases by 1 upon arrival of the new transaction. We have

$$p_1 = \left(\frac{K(t-h)}{2L_0} \right)^2 + 2 \times \frac{K(t-h)}{2L_0} \left(1 - \frac{K(t-h)}{L_0} \right),$$

$$p_2 = \left(\frac{K(t-h)}{2L_0} \right)^2.$$

To obtain the first expression, observe that p_1 equals the probability that both approved tips belong to B plus twice the probability that the first tip belongs to B and the second tip does not belong to $A \cup B$. Analogous to (3), the differential equation for $K(t)$ is:

$$\frac{dK(t)}{dt} = (p_1 - p_2)\lambda = \lambda \frac{K(t-h)}{L_0} \left(1 - \frac{K(t-h)}{L_0} \right). \quad (4)$$

It is difficult to solve (4) exactly, so we make further simplifying assumptions. First of all, we observe that after the time when $K(t)$ reaches level εL_0 for a fixed $\varepsilon > 0$, it will grow very quickly to $(1-\varepsilon)L_0$. Now, when $K(t)$ is small with respect to L_0 , we can drop the last factor in the right-hand side of (4)¹⁷. We obtain a simplified version of (4) by recalling that $\frac{\lambda h}{L_0} = \frac{1}{2}$:

$$\frac{dK(t)}{dt} \approx \frac{1}{2h} K(t-h), \quad (5)$$

with boundary condition $K(0) = 1$. We look for a solution of the form $K(t) = \exp(c\frac{t}{h})$; after substituting this into (5), we obtain

$$\frac{c}{h} \exp\left(c\frac{t}{h}\right) \approx \frac{1}{2h} \exp\left(c\frac{t}{h} - c\right),$$

therefore

$$K(t) = \exp\left(W\left(\frac{1}{2}\right)\frac{t}{h}\right) \approx \exp\left(0.352\frac{t}{h}\right) \quad (6)$$

is an approximate solution, where $W(\cdot)$ is the so-called Lambert W -function.¹⁸ Taking the logarithm of both sides in (6), we find that the time when $K(t)$ reaches εL_0 is

¹⁷It would be a constant close to 1, so the right-hand side would be equivalent to $\lambda \frac{K(t-h)}{L_0}$.

¹⁸Also known as the omega function or product logarithm; for $x \in [0, +\infty)$ it is characterized by the relation $x = W(x) \exp(W(x))$.

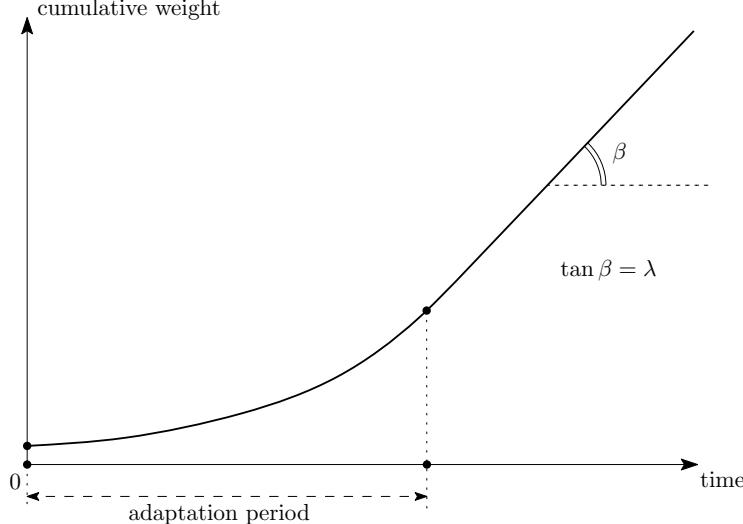


Figure 4: Plot of cumulative weight vs. time for the high load regime.

roughly

$$t_0 \approx \frac{h}{W\left(\frac{1}{2}\right)} \times (\ln L_0 - \ln \varepsilon^{-1}) \lesssim 2.84 \cdot h \ln L_0. \quad (7)$$

Returning to (3) and dropping the last term on the right-hand side, we obtain that during the “adaptation period” (i.e., $t \leq t_0$ with t_0 as in (7)), it holds that

$$\begin{aligned} \frac{dH(t)}{dt} &\approx \frac{2\lambda}{L_0} K(t-h) \\ &\approx \frac{1}{h \exp(W(\frac{1}{2}))} \exp\left(W(\frac{1}{2})\frac{t}{h}\right) \\ &= \frac{2W(\frac{1}{2})}{h} \exp\left(W(\frac{1}{2})\frac{t}{h}\right) \end{aligned}$$

and therefore

$$H(t) \approx 2 \exp\left(W(\frac{1}{2})\frac{t}{h}\right) \approx 2 \exp\left(0.352\frac{t}{h}\right). \quad (8)$$

Let us also remind the reader that after the adaptation period, the cumulative weight $H(t)$ grows linearly with speed λ . We stress that the “exponential growth” in (8) does not mean that the cumulative weight grows “very quickly” during the adaptation period. Rather, the behavior is as depicted in Figure 4.

Conclusions:

1. After a transaction gets approved multiple times in the low load regime, its cumulative weight will grow with speed λw , where w is the mean weight of a generic transaction.
2. In the high load regime, there are two distinct growth phases. First, a transaction's cumulative weight $H(t)$ grows with increasing speed during the *adaptation period* according to (8). After the adaptation period is over, the cumulative weight grows with speed λw (Figure 4). In fact, for *any* reasonable strategy the cumulative weight will grow with this speed after the end of the adaptation period because all incoming transactions will indirectly approve the transaction of interest.
3. One can think of the adaptation period of a transaction as the time until most of the current tips indirectly approve that transaction. The typical length of the adaptation period is given by (7).

4 Possible attack scenarios

We start by discussing an attack scenario where the attacker tries to “outpace” the network alone:

1. An attacker sends a payment to a merchant and receives the goods after the merchant decides the transaction has a sufficiently large cumulative weight.
2. The attacker issues a double-spending transaction.
3. The attacker uses their computing power to issue many small transactions that approve the double-spending transaction, but do not approve the original transaction that they sent to the merchant either directly or indirectly.
4. It is possible for the attacker to have a plethora of Sybil identities which are not required to approve tips.
5. An alternative method to item 3 would be for the attacker to issue a big double-spending transaction using all of their computing power. This transaction would have a very large own weight¹⁹, and would approve transactions prior to the legitimate transaction used to pay the merchant.

¹⁹Here we assume that the own weight of a transaction may vary. It will become clear in the discussion below why it is a good idea to let the own weight vary.

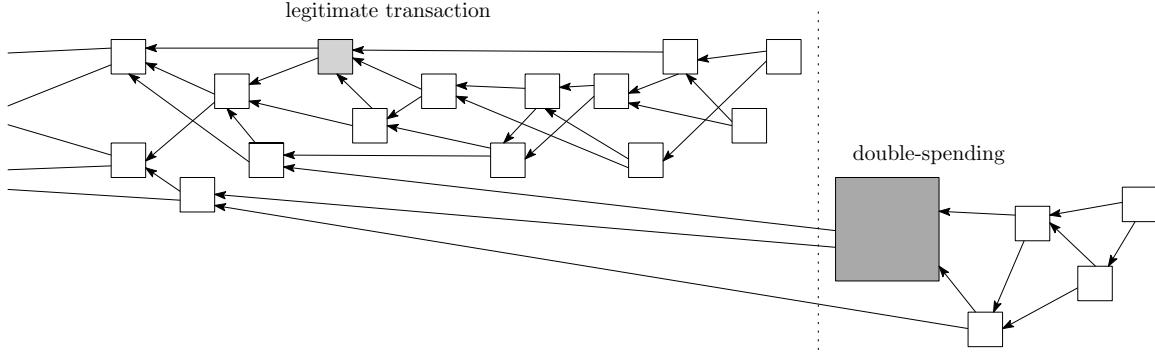


Figure 5: The “large weight” attack

6. The attacker hopes that their dishonest subtangle outpaces the honest subtangle. If this happens, the main tangle continues growing from the double-spending transaction, and the legitimate branch with the original payment to the merchant is orphaned (Figure 5).

In fact, it can be shown that the strategy of one large double-spending transaction increases the attacker’s chances of being successful. In the “ideal” situation of this mathematical model, this attack *always* succeeds.

Let $W^{(n)}$ be the time needed to obtain a nonce that gives the double-spending transaction a weight of at least 3^n . One may assume that $W^{(n)}$ is an exponentially distributed random variable with parameter²⁰ $\mu 3^{-n}$, where μ represents the computing power of the attacker.

Assume that the merchant accepts the legitimate transaction when its cumulative weight becomes at least w_0 , which happens t_0 time units after the original transaction. It is reasonable to expect that the cumulative weight grows with linear speed λw , where λ is the overall arrival rate of transactions issued on the network by honest nodes, and w is the mean weight of a generic transaction. The typical total weight of the legitimate branch at that time is $w_1 = \lambda w t_0$.

Let $\lceil x \rceil$ be the smallest integer greater than or equal to x , define $n_0 = \lceil \frac{\ln w_1}{\ln 3} \rceil$, so that $3^{n_0} \geq w_1$ ²¹. If the attacker managed to obtain a nonce that gives the double-spending transaction a weight of at least 3^{n_0} during the time interval of length t_0 , then the attack succeeds. The probability of this event is

$$\mathbb{P}[W^{(n_0)} < t_0] = 1 - \exp(-t_0 \mu 3^{-n_0}) \approx 1 - \exp(-t_0 \mu w_1^{-1}) \approx \frac{t_0 \mu}{w_1}.$$

²⁰With expectation $\mu^{-1} 3^n$.

²¹In fact, $3^{n_0} \approx w_1$ if w_1 is large.

This approximation is true in the case where $\frac{t_0\mu}{w_1}$ is small, which is a reasonable assumption. If this “immediate” attack does not succeed, the attacker may continue to look for the nonce that gives weight 3^n for $n > n_0$, and hope that at the moment they find it, the total weight of the legitimate branch is smaller than 3^n . The probability of this event occurring is

$$\mathbb{P}[\lambda w W^{(n)} < 3^n] = 1 - \exp(-\mu 3^{-n_0} \times (3^{n_0}/\lambda w)) = 1 - \exp(-\mu/\lambda w) \approx \frac{\mu}{\lambda w}.$$

That is, although $\frac{\mu}{\lambda w}$ should typically be a small number, at each “level” n the attack succeeds with a constant probability. Therefore, it will a.s. succeed. The typical time until it succeeds is roughly $3^{\frac{\lambda w}{\mu}}$. Although this quantity may be very large, the probability that the “first”²² attack succeeds is not negligible. Therefore, we need countermeasures. One such countermeasure would be limiting the own weight from above, or even setting it to a constant value. As mentioned in Section 3, the latter may not be the best solution because it does not offer enough protection from spam.

Now, let us discuss the situation where the maximum own weight is capped at a value of 1, and estimate the probability that the attack succeeds.

Assume that a given transaction gained cumulative weight w_0 in t_0 time units after the moment when it was issued, and that the adaptation period for that transaction is over. In this situation, the transaction’s cumulative weight increases linearly with speed λ . Now, imagine that the attacker wants to double-spend on this transaction. To do so, the attacker secretly prepares the double-spending transaction, and starts generating *nonsense* transactions that approve the double-spending transaction at the time²³ when the *original* transaction was issued to the merchant. If the attacker’s subtangle outpaces the legitimate subtangle at some moment after the merchant decides to accept the legitimate transaction, then the double-spending attack would be successful. If that does not happen, then the double-spending transaction would not be approved by others because the legitimate transaction would acquire more cumulative weight and essentially all new tips would indirectly approve it. The double-spending transaction would be orphaned in this scenario.

As before, let μ stand for the computing power of the attacker. We also make a simplifying assumption that the transactions propagate instantly. Let G_1, G_2, G_3, \dots denote i.i.d. exponential random variables with parameter μ ²⁴, and define $V_k = \mu G_k$, $k \geq 1$. It follows that V_1, V_2, V_3, \dots are i.i.d. exponential random variables with parameter 1.

²²During the time t_0 .

²³Or even before; we discuss this case later.

²⁴With expected value $1/\mu$.

Suppose that at time t_0 the merchant decides to accept the transaction with cumulative weight w_0 . Let us estimate the probability that the attacker successfully double-spends. Let $M(\theta) = (1 - \theta)^{-1}$ be the moment generating function of the exponential distribution with parameter 1 (Section 7.7 of [14]). It is known²⁵ that for $\alpha \in (0, 1)$ it holds that

$$\mathbb{P}\left[\sum_{k=1}^n V_k \leq \alpha n\right] \approx \exp(-n\varphi(\alpha)), \quad (9)$$

where $\varphi(\alpha) = -\ln \alpha + \alpha - 1$ is the Legendre transform of $\ln M(\theta)$. As a general fact, it holds that $\varphi(\alpha) > 0$ for $\alpha \in (0, 1)$. Recall that the expectation of an exponential random variable with parameter 1 also equals 1.

Assume that $\frac{\mu t_0}{w_0} < 1$, otherwise the probability that the attacker's subtangle eventually outpaces the legitimate subtangle would be close to 1. Now, to outweigh w_0 at time t_0 , the attacker needs to be able to issue at least w_0 transactions with maximum own weight m during time t_0 . Therefore, using (9), we find the probability that the double-spending transaction has more cumulative weight at time t_0 is roughly

$$\begin{aligned} \mathbb{P}\left[\sum_{k=1}^{w_0/m} G_k < t_0\right] &= \mathbb{P}\left[\sum_{k=1}^{w_0} V_k < \mu t_0\right] \\ &= \mathbb{P}\left[\sum_{k=1}^{w_0} V_k < w_0 \times \frac{\mu t_0}{w_0}\right] \\ &\approx \exp(-w_0\varphi(\frac{\mu t_0}{w_0})). \end{aligned} \quad (10)$$

For the above probability to be small, $\frac{w_0}{m}$ needs to be large and $\varphi(\frac{\mu t_0}{w_0})$ cannot be very small.

Note that, at time $t \geq t_0$, the cumulative weight of the legitimate transaction is roughly $w_0 + \lambda(t - t_0)$ because we assumed that the adaptation period is over, so the cumulative weight grows with speed λ . Analogous to (10), one finds the probability that the double-spending transaction has more cumulative weight at time $t \geq t_0$ is roughly

$$\exp(-(w_0 + \lambda(t - t_0))\varphi(\frac{\mu t}{w_0 + \lambda(t - t_0)})). \quad (11)$$

Then, it must be true that we have $\frac{\mu t_0}{w_0} \geq \frac{\mu}{\lambda}$ since the cumulative weight grows with speed less than λ during the adaptation period. It can be shown that the probability

²⁵This is a consequence of the so-called Large Deviation Principle. See the general book [13], and Proposition 5.2 in Section 8.5 of [14] for a simple and instructive derivation of the upper bound, and Section 1.9 of [5] for the (not so simple) derivation of the lower bound.

of achieving a successful double spend is of order

$$\exp\left(-w_0\varphi\left(\max\left(\frac{\mu t_0}{w_0}, \frac{\mu}{\lambda}\right)\right)\right). \quad (12)$$

For example, let $\mu = 2$, $\lambda = 3$ so that the attacker's power is only a bit less than that of the rest of the network. Assume that the transaction has a cumulative weight of 32 by time 12. Then, $\max\left(\frac{\mu t_0}{w_0}, \frac{\mu}{\lambda}\right) = \frac{3}{4}$, $\varphi\left(\frac{3}{4}\right) \approx 0.03768$, and (12) then gives the upper bound approximately 0.29. If one assumes that $\mu = 1$ and keeps all other parameters intact, then $\max\left(\frac{\mu t_0}{w_0}, \frac{\mu}{\lambda}\right) = \frac{3}{8}$, $\varphi\left(\frac{3}{8}\right) \approx 0.3558$, and (12) gives approximately 0.00001135, quite a drastic change.

From the above discussion it is important to recognize that the inequality $\lambda > \mu$ should be true for the system to be secure. In other words, the input flow of "honest" transactions should be large compared to the attacker's computational power. Otherwise, the estimate (12) would be useless. This indicates the need for additional security measures, such as checkpoints, during the early days of a tangle-based system.

When choosing a strategy for deciding which one of two conflicting transactions is valid, one has to be careful when using cumulative weight as a decision metric. This is due to the fact that cumulative weight can be subject to an attack similar to the one described in Section 4.1, namely the attacker may prepare a double-spending transaction well in advance, build a secret subtangle referencing it, and then broadcast that subtangle after the merchant accepts the legitimate transaction. A better method for deciding between two conflicting transactions might be the one described in the next section: run the tip selection algorithm and see which of the two transactions is indirectly approved by the selected tip.

4.1 A parasite chain attack and a new tip selection algorithm

Consider the following attack (Figure 6): an attacker secretly builds a subtangle that occasionally references the main tangle to gain a higher score. Note that the score of honest tips is roughly the sum of all own weights in the main tangle, while the score of the attacker's tips also contains the sum of all own weights in the parasite chain. Since network latency is not an issue for an attacker who builds a subtangle alone²⁶, they might be able to give more height to the parasite tips if they use a computer that is sufficiently strong. Moreover, the attacker can artificially increase their tip count at the moment of the attack by broadcasting many new transactions

²⁶This is due to the fact that an attacker can always approve their own transactions without relying on any information from the rest of the network.

that approve transactions that they issued earlier on the parasite chain (Figure 6). This will give the attacker an advantage in the case where the honest nodes use some selection strategy that involves a simple choice between available tips.

To defend against this attack style, we are going to use the fact that the main tangle is supposed to have more active hashing power than the attacker. Therefore, the main tangle is able to produce larger increases in cumulative weight for more transactions than the attacker. The idea is to use a MCMC algorithm to select the two tips to reference.

Let \mathcal{H}_x be the current cumulative weight of a site. Recall that we assumed all own weights are equal to 1. Therefore, the cumulative weight of a tip is always 1, and the cumulative weight of other sites is at least 2.

The idea is to place some particles, a.k.a. random walkers, on sites of the tangle and let them walk towards the tips in a random²⁷ way. The tips “chosen” by the walks are then the candidates for approval. The algorithm is described in the following way:

1. Consider all sites on the interval $[W, 2W]$, where W is reasonably large²⁸.
2. Independently place N particles on sites in that interval²⁹.
3. Let these particles perform independent discrete-time random walks “towards the tips”, meaning that a transition from x to y is possible if and only if y approves x
4. The two random walkers that reach the tip set first will sit on the two tips that will be approved. However, it may be wise to modify this rule in the following way: first discard those random walkers that reached the tips *too fast* because they may have ended on one of the “lazy tips”.
5. The transition probabilities of the walkers are defined in the following way: if y approves x ($y \rightsquigarrow x$), then the transition probability P_{xy} is proportional to

²⁷There is not a “canonical” source of randomness. The nodes just use their own (pseudo)random number generators to simulate the random walks.

²⁸The idea is to place the particle “deep” into the tangle so that it will not arrive at a tip straight away. However, the particle should not be placed “too deep” because it needs to find a tip in a reasonable time. Also, the interval $[W, 2W]$ is arbitrary. One could chose $[W, 5W]$, etc. There are also other ways to select the walkers’ starting points. For example, a node can simply take a random transaction received between t_0 and $2t_0$ time units in the past, where t_0 is some fixed time point.

²⁹This choice is largely arbitrary. We use several particles instead of just two for additional security. The idea is that if a particle were to accidentally jump to the attacker’s chain, which is supposed to be long, then it would spend a lot of time there and other tips will be chosen first.

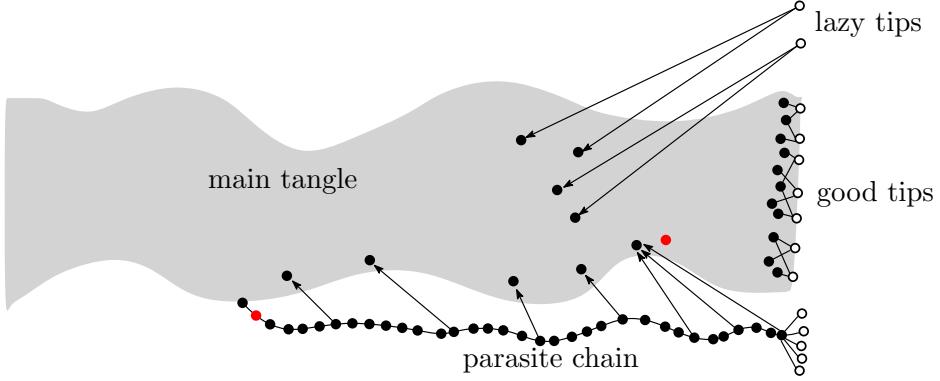


Figure 6: Visual representation of the tip selection algorithm for honest tips, as well as the parasite chain. The two red circles indicate an attempted double-spend by an attacker.

$\exp(-\alpha(\mathcal{H}_x - \mathcal{H}_y))$, that is

$$P_{xy} = \exp(-\alpha(\mathcal{H}_x - \mathcal{H}_y)) \left(\sum_{z:z \rightsquigarrow x} \exp(-\alpha(\mathcal{H}_x - \mathcal{H}_z)) \right)^{-1}, \quad (13)$$

where $\alpha > 0$ is a parameter to be chosen³⁰.

Note that this algorithm is “local”, meaning one does not need to traverse the tangle back to the genesis to perform relevant calculations. In particular, observe that one does not need to calculate the cumulative weights for the whole tangle. At most one needs to calculate the cumulative weights for the sites that indirectly approve the starting point of the walker.

To check that the algorithm works as intended, first consider the “lazy tips”. These tips intentionally approve some old transactions to avoid doing verification work (Figure 6). Even if the particle is on a site approved by a lazy tip, it is not probable that the lazy tip would be selected because the difference between cumulative weights would be very large and P_{xy} would be small.

Next, consider this alternate attack style: the attacker secretly builds a chain containing a transaction that empties their account balance to another account under their control, indicated as the leftmost red circle in Figure 6. Then, the attacker issues a transaction on the main tangle, represented by the rightmost red circle, and waits for the merchant to accept it. The parasite chain occasionally references the main

³⁰One can start with $\alpha = 1$.

tangle. However, the cumulative weight is not very large in the parasite chain. It should be noted that the parasite chain cannot reference the main tangle after the merchant’s transaction. Furthermore, the attacker might try to artificially inflate the number of tips in their parasite chain at the moment of the attack (Figure 6). The attacker’s idea is to make the nodes issuing new transactions reference the parasite chain so that the honest branch of the tangle will be orphaned.

It is easy to see why the MCMC selection algorithm will not select one of the attacker’s tips with high probability. The reasoning is identical to the lazy tip scenario: the sites on the parasite chain will have a cumulative weight that is much smaller than the sites that they reference on the main tangle. Therefore, it is not probable that the random walker will ever jump to the parasite chain unless it begins there, and this event is not very probable either because the main tangle contains more sites.

As an additional protecting measure, we can first run a random walk with a large α (so that it is in fact “almost deterministic”) to choose a “model tip”; then, use random walks with small α for actual tip selection, but verify if the (indirectly) referenced transactions are consistent with the model tip.

Observe also that, for a random walk that *always* moves towards the tips it is very simple and rapid to calculate the exit probability distribution using a straightforward recursion; this is something that we *do not* want the nodes to do. However, it is possible to modify our approach in the following way: on each step, the random walk may backtrack (i.e., go 1 step away from the tips) with probability (say) $\frac{1}{3}$ (and divide the remaining $\frac{2}{3}$ as before). The walk will reach the tips very quickly anyway (because it has a drift towards the tips), but it will not be so easy to calculate the exit measure.

Let us comment on why the nodes would follow this algorithm. Recall from Section 1 that it is reasonable to assume that at least a “good” proportion of the nodes will follow the *reference* algorithm. Also, because of computational and network delays, the tip selection algorithm would rather work with a past snapshot of the tangle with respect to the moment when a transaction is issued. *It may be a good idea to intentionally move this snapshot to a time point further in the past*³¹ in the reference algorithm for the reasons that we explain in the sequel. Imagine a “selfish” node that just wants to maximize the chances of their transaction being approved quickly. The MCMC algorithm of this section, which is adopted by a considerable proportion of nodes, defines a probability distribution on the set of tips. It is clear that

³¹First the random walker finds a former tip with respect to that snapshot, and then it continues to walk towards the “actual” tips on the current tangle.

a natural first choice for a selfish node would be to choose the tips where the maximum of that distribution is attained. However, if many other nodes also behave in a selfish way and use the same strategy, which is a reasonable assumption, then they all will lose. *Many* new transactions will approve the same two tips at roughly the same time, therefore generating too much competition between them for subsequent approval. It should also be clear that nodes will not immediately “feel” the cumulative weight increase caused by this mass approval of the same two tips since the nodes are using a past snapshot. For this reason, even a selfish node would have to use some random tip approval algorithm³² with a probability distribution for tip selection that is close to the default probability distribution produced by the reference tip selection algorithm. We do not claim that this “aggregated” probability distribution would be equal to the default probability distribution in the presence of selfish nodes. However, the above argument shows that it should be close to it. This means that the probability of many nodes attempting to verify the same “bad” tips would remain small. In any case, there is not a large incentive for the nodes to be selfish because possible gains only amount to a slight decrease in confirmation time. This is inherently different from other decentralized constructs, such as Bitcoin. The important fact is that nodes do not have reasons to abandon the MCMC tip selection algorithm.

We would like to mention that the definition of transition probabilities, as given in (13), has not been set in stone. Instead of the exponent, one can use a different function that decreases rapidly, such $f(s) = s^{-3}$. There is also freedom for choosing W and N as well. At this point in time, it is unclear if there are any theoretical arguments that show exactly in which way these parameters should be chosen. In sum, we feel that the main contribution of this section is the idea of using MCMC for tip selection.

4.2 Splitting attack

Aviv Zohar suggested the following attack scheme against the proposed MCMC algorithm. In the high-load regime, an attacker can try to split the tangle into two branches and maintain the balance between them. This would allow both branches to continue to grow. The attacker must place at least two conflicting transactions

³²as noticed before, for a backtracking walk there seem to be no easy way to discover which tips are better (that is, more likely to be selected by “honest” nodes) other than running the MCMC many times. However, running MCMC many times requires time and other resources; after one spends some time on it, the state of the tangle will already change, so one would possibly even have to start anew. This explains why nodes do not have reasons to abandon the MCMC tips selection strategy in favor of something else, at least if they assume that a considerable proportion of the other nodes follow the default tips selection strategy.

at the beginning of the split to prevent an honest node from effectively joining the branches by referencing them both simultaneously. Then, the attacker hopes that roughly half of the network would contribute to each branch so that they would be able to “compensate” for random fluctuations, even with a relatively small amount of personal computing power. If this technique works, the attacker would be able to spend the same funds on the two branches.

To defend against such an attack, one needs to use a “sharp-threshold” rule that makes it too hard to maintain the balance between the two branches. An example of such a rule is selecting the longest chain on the Bitcoin network. Let us translate this concept to the tangle when it is undergoing a splitting attack. Assume that the first branch has total weight 537, and the second branch has total weight 528. If an honest node selects the first branch with probability very close to $1/2$, then the attacker would probably be able to maintain the balance between the branches. However, if an honest node selects the first branch with probability much larger than $1/2$, then the attacker would probably be unable to maintain the balance. The inability to maintain balance between the two branches in the latter case is due to the fact that after an inevitable random fluctuation, the network will quickly choose one of the branches and abandon the other. In order to make the MCMC algorithm behave this way, one has to choose a very rapidly decaying function f , and initiate the random walk at a node with large depth so that it is highly probable that the walk starts before the branch bifurcation. In this case, the random walk would choose the “heavier” branch with high probability, even if the difference in cumulative weight between the competing branches is small.

It is worth noting that the attacker’s task is very difficult because of network synchronization issues: they may not be aware of a large number of recently issued transactions³³. Another effective method for defending against a splitting attack would be for a sufficiently powerful entity to instantaneously publish a large number of transactions on one branch, thus rapidly changing the power balance and making it difficult for the attacker to deal with this change. If the attacker manages to maintain the split, the most recent transactions will only have around 50% confirmation confidence (Section 1), and the branches will not grow. In this scenario, the “honest” nodes may decide to start selectively giving their approval to the transactions that occurred before the bifurcation, bypassing the opportunity to approve the conflicting transactions on the split branches.

One may consider other versions of the tip selection algorithm. For example, if a node sees two big subtangles, then it chooses the one with a larger sum of own

³³The “real” cumulative weights may be quite different from what they believe.

weights before performing the MCMC tip selection algorithm outlined above.

The following idea may be worth considering for future implementations. One could make the transition probabilities defined in (13) depend on both $\mathcal{H}_x - \mathcal{H}_y$ and \mathcal{H}_x in such a way that the next step of the Markov chain is almost deterministic when the walker is deep in the tangle, yet becomes more random when the walker is close to tips. This will help avoid entering the weaker branch while assuring sufficient randomness when choosing the two tips to approve.

Conclusions:

1. We considered attack strategies for when an attacker tries to double-spend by “outpacing” the system.
2. The “large weight” attack means that, in order to double-spend, the attacker tries to give a very large weight to the double-spending transaction so that it would outweigh the legitimate subtangle. This strategy would be a menace to the network in the case where the allowed own weight is unbounded. As a solution, we may limit the own weight of a transaction from above, or set it to a constant value.
3. In the situation where the maximal own weight of a transaction is m , the best attack strategy is to generate transactions with own weight m that reference the double-spending transaction. When the input flow of “honest” transactions is large enough compared to the attacker’s computational power, the probability that the double-spending transaction has a larger cumulative weight can be estimated using the formula (12) (see also examples below (12)).
4. The attack method of building a “parasite chain” makes approval strategies based on height or score obsolete since the attacker’s sites will have higher values for these metrics when compared to the legitimate tangle. On the other hand, the MCMC tip selection algorithm described in Section 4.1 seems to provide protection against this kind of attack.
5. The MCMC tip selection algorithm also offers protection against the lazy nodes as a bonus.

5 Resistance to quantum computations

It is known that a sufficiently large quantum computer³⁴ could be very efficient for handling problems that rely on trial and error to find a solution. The process of finding a nonce in order to generate a Bitcoin block is a good example of such a problem. As of today, one must check an average of 2^{68} nonces to find a suitable hash that allows a new block to be generated. It is known (see e.g. [15]) that a quantum computer would need $\Theta(\sqrt{N})$ operations to solve a problem that is analogous to the Bitcoin puzzle stated above. This same problem would need $\Theta(N)$ operations on a classical computer. Therefore, a quantum computer would be around $\sqrt{2^{68}} = 2^{34} \approx 17$ billion times more efficient at mining the Bitcoin blockchain than a classical computer. Also, it is worth noting that if a blockchain does not increase its difficulty in response to increased hashing power, there would be an increased rate of orphaned blocks.

For the same reason, a “large weight” attack would also be much more efficient on a quantum computer. However, capping the weight from above, as suggested in Section 4, would effectively prevent a quantum computer attack as well. This is evident in iota because the number of nonces that one needs to check in order to find a suitable hash for issuing a transaction is not unreasonably large. On average, it is around 3^8 . The gain of efficiency for an “ideal” quantum computer would therefore be of order $3^4 = 81$, which is already quite acceptable³⁵. More importantly, the algorithm used in the iota implementation is structured such that the time to find a nonce is not much larger than the time needed for other tasks that are necessary to issue a transaction. The latter part is much more resistant against quantum computing, and therefore gives the tangle much more protection against an adversary with a quantum computer when compared to the (Bitcoin) blockchain.

Acknowledgements

The author thanks Bartosz Kusmierz, Cyril Grünspan, Olivia Saa, Razvan Savu, Samuel Reid, Toru Kazama, Rafael Kallis, and Rodrigo Bueno who pointed out several errors in earlier drafts, and James Brogan for his contributions towards making this paper more readable.

³⁴Still a hypothetical construct as of today.

³⁵Note that $\Theta(\sqrt{N})$ could easily mean $10\sqrt{N}$.

References

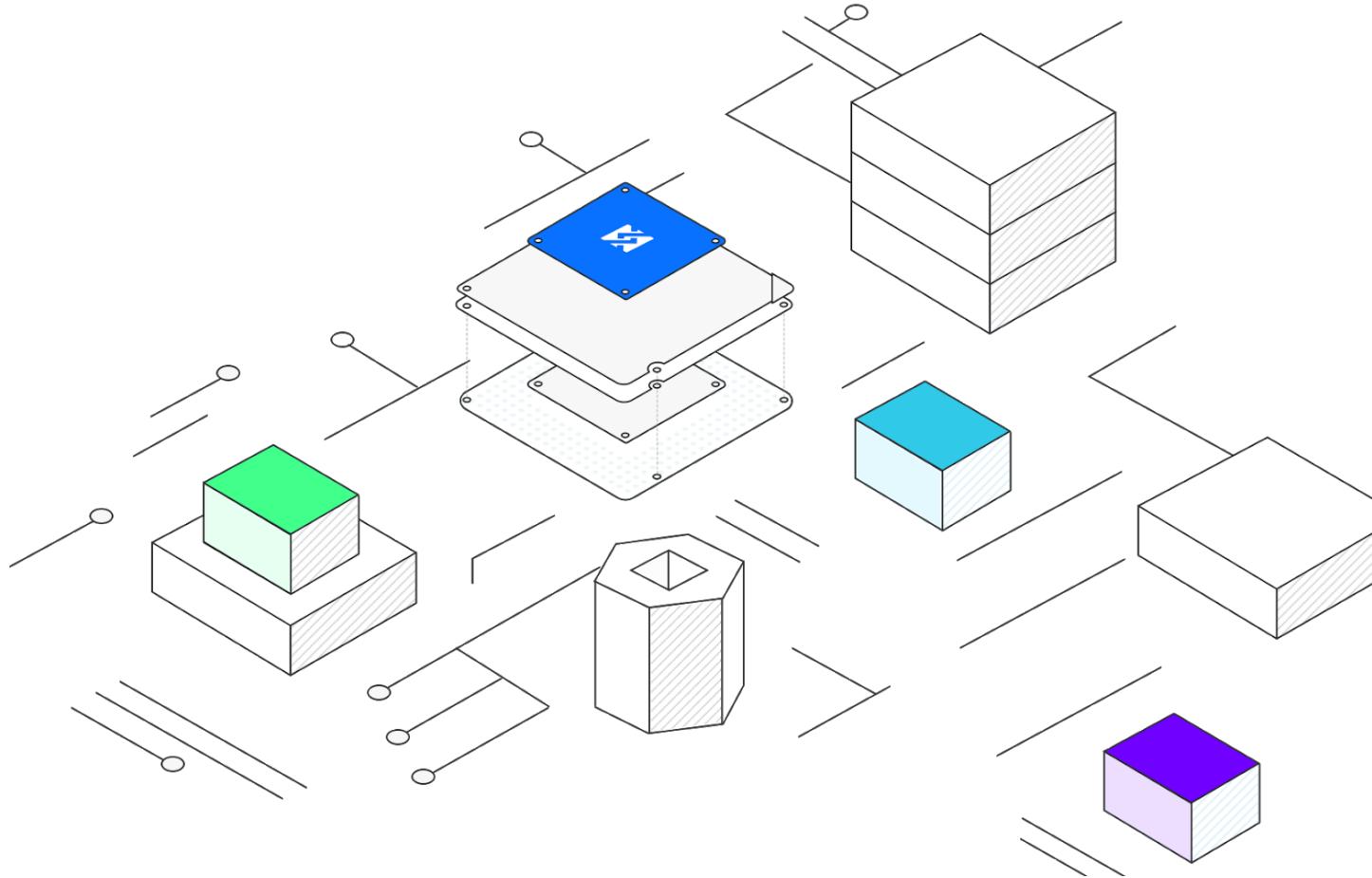
- [1] Iota: a cryptocurrency for Internet-of-Things. See <http://www.iotatoken.com/>, and <https://bitcointalk.org/index.php?topic=1216479.0>
- [2] bitcoinj. Working with micropayment channels.
<https://bitcoinj.github.io/working-with-micropayments>
- [3] PEOPLE ON NXTFORUM.ORG (2014) DAG, a generalized blockchain.
<https://nxtforum.org/proof-of-stake-algorithm/dag-a-generalized-blockchain/> (registration at nxtforum.org required)
- [4] MOSHE BABAIOFF, SHAHAR DOBZINSKI, SIGAL OREN, AVIV ZOHAR (2012) On Bitcoin and red balloons. *Proc. 13th ACM Conf. Electronic Commerce*, 56–73.
- [5] RICHARD DURRETT (2004) Probability – Theory and Examples. *Duxbury advanced series*.
- [6] SERGIO DEMIAN LERNER (2015) DagCoin: a cryptocurrency without blocks.
<https://bitslog.wordpress.com/2015/09/11/dagcoin/>
- [7] YONATAN SOMPOLINSKY, AVIV ZOHAR (2013) Accelerating Bitcoin’s Transaction Processing. Fast Money Grows on Trees, Not Chains.
<https://eprint.iacr.org/2013/881.pdf>
- [8] YONATAN SOMPOLINSKY, YOAD LEWENBERG, AVIV ZOHAR (2016) SPECTRE: Serialization of Proof-of-work Events: Confirming Transactions via Recursive Elections. <https://eprint.iacr.org/2016/1159.pdf>
- [9] YOAD LEWENBERG, YONATAN SOMPOLINSKY, AVIV ZOHAR (2015) Inclusive Block Chain Protocols.
http://www.cs.huji.ac.il/~avivz/pubs/15/inclusive_btc.pdf
- [10] JOSEPH POON, THADDEUS DRYJA (2016) The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments.
<https://lightning.network/lightning-network-paper.pdf>
- [11] SHELDON M. ROSS (2012) *Introduction to Probability Models*. 10th ed.
- [12] DAVID VORICK (2015) Getting rid of blocks. slides.com/davidvorick/braids

- [13] AMIR DEMBO, OFER ZEITOUNI (2010) *Large Deviations Techniques and Applications*. Springer.
- [14] SHELDON M. ROSS (2009) *A First Course in Probability*. 8th ed.
- [15] GILLES BRASSARD, PETER HYER, ALAIN TAPP (1998) Quantum cryptanalysis of hash and claw-free functions. *Lecture Notes in Computer Science* **1380**, 163–169.



Oxcert protocol

Whitepaper 0.5 (draft)



Authors: Kristijan Sedlak, Jure Zih, Mitja Pirc, Urban Osvald

Date: December 31th, 2018

Version: Draft Version 0.5¹

¹ This version and its contents are current as of 2018-07-05 and supersede all previous versions of this Whitepaper or any public statements made about Oxcert and the ZXC Token Sale and are subject to change. This English version is to be relied upon as the most accurate and updated as other language translations may have mistranslations and be outdated.

Abstract

Create, own, and validate unique assets on the blockchain with Oxcert - **the first open protocol** built to support the future of digital assets, powered by non-fungible tokens.

The Oxcert protocol offers **tools for building powerful dapps**, aimed at easy authentication and management of digital or real-world tangible assets (such as ID, educational certificate, in-game item or a house) on the blockchain. In addition to common functions for transferring and managing standard non-fungible tokens, the Oxcert protocol provides **another layer of conventions for creating certified non-fungible tokens for unique assets**. These tokens are called Xcerts and are created through a custom minting process. Xcerts represent opinionated non-fungible tokens, which also hold an imprint of an asset. With Oxcert protocol, we can validate a proof of existence, authenticity and ownership of these digital assets without third-party involvement.

Due to the complexity of low-level blockchain solutions, the wide adoption of non-fungible tokens and blockchain in general is slower than it could be. The lack of conventions prevents interoperability among applications. Developers trying to develop their own decentralised application using non-fungible tokens face long development time and huge risk in their development process and security, lowering the overall efficiency and adoption rate. The resulting ecosystem of digital assets is under risk of being fragmented, with non-interoperable dapps and underlying data. One of the major problems in the future may not be the technological barrier to issuing ownership rights of different unique assets on the blockchain, but rather the authenticity of issuing entities. Currently there is no mechanism to attest credible NFT issuers.

The vision of Oxcert is to provide an **open protocol for standardized and certified non-fungible tokens to a wider tech audience**. With Oxcert, you can build on top of the non-fungible token standard, employing a **complete toolset, development framework, and a set of conventions for various use cases**. This results in shorter development time, lower risk and cuts cost associated with developing blockchain solutions. A wide range of decentralized applications and business models can be supported, giving companies the power to fully utilize the potential of blockchain technology. Oxcert is an

open source protocol that translates one-of-a-kind digital or real-world assets into non-fungible tokens (NFTs) - unique proof of ownership available from the blockchain.

Oxcert is a framework with a set of on-chain and off-chain rules for managing Xcerts and other standard non-fungible tokens. Our mission is to equip application developers with a secure blockchain agnostic platform, powerful tools and community embraced conventions for managing non-fungible tokens. Oxcert is a pluggable settlement with an advanced integration layer for different dapps and relay applications. This enables developers to focus on the application layer and quickly build applications for issuing university certificates, KYC applications, applications for loyalty programs, warranties, badges, credits or even a decentralized non-fungible exchange.

A key role in the Oxcert infrastructure is played by the ZXC utility tokens. These are fungible tokens that are compliant with Ethereum's ERC-20 standard. The ZXC token will be utilized to support dapps built on top of the Oxcert protocol with minimum possible fees. Since Oxcert is an open source project that strives to be community driven, a decentralized governance model could be introduced as well.

We recognize that one of the fundamental problems in the blockchain space may become the authenticity of issuers in the future. A decentralized issuer verification registry may be an acceptable and self-sustaining solution to the problem. The community would have an option to either verify or reject new Xcert issuers through a staking and rewarding mechanism based on the ZXC token.

Furthermore, Oxcert is also building out a whole ecosystem of parties involved in the non-fungible space, as well as specific application developers, companies from various verticals, researchers, organisations and communities. Planned growth activities will not only positively impact and extend further adoption of technologies developed by Oxcert, but also expand the scientific horizon of non-fungible tokens in general.

Disclaimer

This Whitepaper has been issued by Oxcert d.o.o. (the "Company") and should be read in conjunction with the Company's terms and conditions (the "Terms").

The purpose of this Whitepaper is to provide prospective purchasers with the information on the Company's project to allow the prospective purchasers to make their own decision as to whether or not they wish to proceed to purchase a ZXC token. This Whitepaper does not constitute an offer or invitation, sale or purchase of shares, securities or any of the assets of the Company.

As of the date of this Whitepaper, the information contained herein is accurate to the best of the management team's knowledge and there are no other facts of omission, which would make any misleading statements in this Whitepaper. No representation, warranty, assurance or undertaking is made as to its continued accuracy after such date. The information contained in this Whitepaper may be subject to modification, supplementation and amendment at any time moving forward and would be documented accordingly. In addition new information might be added in the future.

This Whitepaper describes the Company's business objectives and the issue by the Company of ZXC tokens. It has not been reviewed, verified, approved or authorized by any regulatory or supervisory authority.

The publication of this Whitepaper and the offering of ZXC tokens may be restricted in certain jurisdictions. It is the responsibility of any person in possession of this Whitepaper and any persons wishing to make an application for ZXC tokens (pursuant to the Terms) to inform themselves of and to observe any and all laws and regulations that may be applicable to them.

This Whitepaper does not constitute an offer or solicitation to anyone in any jurisdiction in which such offer or solicitation is not lawful or in which the person making such offer or solicitation is not qualified to do so.

Prospective purchasers of ZXC tokens should inform themselves as to the legal requirements and consequences of purchasing, holding and disposing of ZXC tokens and any applicable exchange control regulations and taxes in the countries of their respective citizenship, residence and/or domicile.

Prospective purchasers of ZXC tokens are wholly responsible for ensuring that all aspects of this Whitepaper and the Terms are acceptable to them. The purchase of ZXC tokens may involve special risks that could lead to a loss of a substantial portion or a loss of the entire purchase amount. The purchase of ZXC tokens is considered speculative in nature and it involves a high degree of risk. The Company does not represent, warrant, undertake or assure that the ZXC tokens are defect/virus free or

will meet any specific requirements of a prospective purchaser. You should only purchase ZXC tokens if you can afford a complete loss. Unless you fully understand and accept the nature of and the potential risks inherent in the purchase of ZXC tokens, you should not purchase ZXC tokens.

The purchase of ZXC tokens is only possible after the prospective purchaser has read, understood and accepted the Terms. Each prospective purchaser will be required to acknowledge that they made an independent decision to purchase the ZXC tokens and that they are not relying, in any manner whatsoever, on the Company, the management team or any other person or entity (other than such purchaser's own advisers). Prospective purchasers are urged to consult their own legal, tax or other advisor before purchasing ZXC tokens.

The Company and the management team do not provide any advice or recommendations with respect to the ZXC tokens, endorse such tokens, nor do they accept any responsibility or liability for any use of this Whitepaper by any person which is in breach of any local regulatory requirements with regard to the distribution of this Whitepaper or any applicable rules pertaining to the offer of ZXC tokens.

To the maximum extent permitted by the applicable laws, regulations and rules, the Company, its founders, team members and any third parties involved in the Company's project shall not be liable for any indirect, special, incidental, consequential or other losses of any kind. Furthermore, in tort, contract or otherwise (including but not limited to loss of revenue, income or profits and loss of use or data), arising out of or in connection with any acceptance of or reliance on this Whitepaper.

All statements regarding the Company's financial position, business strategies, plans and prospects of the industry which the Company is in are forward looking statements. Neither the Company, its founders, team members or any third parties involved in the Company's project nor any other persons represent, warrant, undertake that the actual future results, performance or achievements of the Company will be as discussed in these forward looking statements.

This Whitepaper includes market and industry information and forecasts, which the Company obtained from internal surveys, reports and studies where appropriate, as well as market research, publicly available information and industry publications. Such surveys, reports, studies, market research, publicly available information and publications state that the information that they contain has come from sources

deemed reliable; there is no assurance as to the accuracy or completeness of such included information.

The Company does not make or purport to make any disclaims, any representation, warranty or undertaking in any form whatsoever to any entity or person. Including any representation, warranty or undertaking about the truth, accuracy, and completeness of any of the information set out in this Whitepaper.

Statements made in this Whitepaper are based on the law and practice currently in force in Slovenia, which is a country in the European Union and are subject to changes in accordance with said laws.

Table of Content

Abstract	1
Disclaimer	2
Table of Content	6
1. Introduction and Vision	8
2. Opportunity	10
3. Solution	12
4. Oxcert Protocol	15
4.1 Overview	15
4.1.1 Fungibility	17
4.1.2 Decentralization	18
4.2 Specification	19
4.3. Xcert	20
4.3 Framework	21
4.3.1 Devkit (SDK)	22
4.3.2 Decentralized exchange	22
4.3.3 Decentralized minter	23
4.3.4 Continuous integration	23
5. Use cases	24
5.1 KYC	24
5.2 Academic credentials	25
5.3 Art	26
5.4 Collectibles	27
6. Ecosystem	29
6.1 Participants	31
6.2 Promotion & Growth	31
6.3 Partnerships	31
6.4 Ecosystem Growth Pool	34
7. Token economy	35
7.1 Token purpose and Use cases	35
7.1.1 ZXC within dapps	35
	6

7.1.2 Decentralized governance	36
7.1.3 Issuer Verification Registry (IVR)	36
7.2 Implementation	37
8. Structure	38
8.1 Corporation Structure	38
8.2 Governance	38
8.3 Token	39
8.4 Exchanges	40
8.5 Voting	40
9. The business side of things	41
9.1 Business model	41
9.2 Go-to-market strategy	41
9.3 Competitive landscape	42
10. Timeline	43
10.1 Development roadmap	43
10.2 Milestones	44
11. Token distribution event	45
11.1 Token and crowdsale info	45
11.2 Token distribution	46
11.3 Distribution schedule	47
11.4 Participation in the token sale	48
11.5 Funds allocation	49
12. Team	51
12.1 Team	51
11.2 Advisors	52
Disclaimer	54

1. Introduction and Vision

We are living in one of the most exciting times since the beginning of the Internet. The advent of blockchain has redefined technology and set the course for the future of everything. Banking, insurance, advertising as well as many other industries are adopting this new paradigm of governance through a new concept introduced by Satoshi Nakamoto back in 2007.

Although blockchain's utility has expanded to numerous use-cases, the core value of the technology remains. It is a tamper-proof, transparent and secure decentralized ledger that maintains a list of records that cannot be altered retroactively. Blockchain was first built for the financial capital; now it is ready to take on social and professional capital: achievements, qualifications, accreditations, credentials and certification.

But the really exciting paradigm shift comes with blockchain's ability to tap into the physical world. With the introduction of unique decentralized assets presented as non-fungible tokens², the idea of storing intangible assets such as copyrights, patents and goodwill, as well as tangible assets such as property, equipment and inventory is now possible like never before.

With the emergence of different business opportunities related to the tokenization of physical assets, there is a **growing need for standardization and a base protocol that would allow for simple proof-validation of its existence, authenticity and ownership.** Furthermore, the future of a new technology of this magnitude should be made available to everyone.

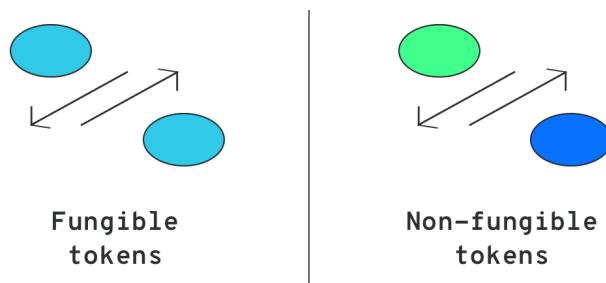
The vision of Oxcert is to provide an open protocol for non-fungible tokens to a wider tech audience. With its help, we can drastically shorten development time, decrease risk and cut costs associated with developing blockchain solutions. A wide range of decentralized applications (dapps) and business models can be supported, giving companies the power to fully utilize the potential of blockchain technology. Non-blockchain companies will be able to make use of a decentralized and distributed ledger to easily incorporate the non-fungible technological features into their applications while not requiring their development team to be proficient in low-level blockchain programming.

² E.g. ERC721 standard on the Ethereum blockchain.

Blockchain is moving faster than anyone dared to predict. With the Oxcert protocol, we are opening the door for all in this important time in history, where everything you own, not just financial assets, will now have a way to exist securely in your digital wallet.

2. Opportunity

The most common tokens of today's crypto economy follow the Ethereum's ERC-20 specification. These tokens are used for a specifically defined utility within predefined systems and thus carry value. Tokens that are issued by the same source are identical and mutually interchangeable. This characteristic is called fungibility, hence these tokens are called fungible tokens. Token holders can buy and sell these tokens on exchanges, which is also the primary mechanism of their price valuation.



Recently, another form of tokens, called **non-fungible tokens (NFTs)**, started getting attention in the crypto community. Non-fungible digital assets represent the next stage in the blockchain evolution. The first good use case for NFTs was introduced with CryptoKitties. These were one of the first popular crypto-collectables: assets that were unique and could be stored in your Ethereum wallet. You could buy, sell, trade and even breed them. Later on the Ethereum implementation proposal 721 (EIP-721) got introduced and confirmed got confirmed as an Ethereum standard (ERC-721) in March 2018.

NFTs have the potential to improve many applications and enhance existing business models. **For the first time, users are actually able to hold distinguishable tokens that carry not only value, but also unique information in their blockchain wallet.** Numerous companies from different verticals are looking into this new technology as a solution for what was previously not possible on the blockchain. According to OpenSea³, three new companies a day are starting to do an implementation with the NFTs. NFT use cases range from identities (KYC), collectibles, to education certificates and more.

³ Source: Keynote from OpenSea, 3/23/2018 at the event Explore 721 Dallas

However, there are four major challenges we see that hinder the development and adoption of NFTs as well as enabling greater technological and business impact:

1. **Low speed of development:** due to the lack of a pluggable framework, the development of dapps can take months therefore severely increasing go-to-market time, which results in much slower adoption and increases competitive risks.
2. **Lack of conventions and interoperability:** different verticals require different conventions which are currently missing to ensure interoperability. Equally important is the gap of interoperability between applications, which needs to be addressed in order to maximize adoption and business impact.
3. **Missing registry of verified issuers:** a big missing part is a registry of issuers of NFTs, which would serve for authentication and validity of the issuers. Creating a common registry will reduce verification time as well as reduce business risks.
4. **Limited open-source and blockchain-agnostic solutions:** current NFT applications are siloed for specific purposes and currently Ethereum-focused, but with opportunities emerging in other blockchains (EOS, NEO) too. The opportunity to have an open source and blockchain agnostic protocol will open far greater opportunities for dapps and usage of NFTs.

Furthermore, there is a need for a unifying technological layer connecting these assets through a higher-level standardization model. Building on top of the existing low-level technology would mean that each development team builds their own framework from scratch. The work gets repeated every time and each solution has to perform a separate audit. There are currently no processes and rules in place that would allow for a faster, secure and interoperable issuance and verification of these assets, consequently hindering the potential for wider adoption.

2018 is frequently called the year of NFTs and the timing for the Oxcert protocol is now. With the ERC-721 in place, we believe that there is a solid foundation for NFTs, resulting in an increased demand for a quickly deployable and effective development environment. We believe that the need for the Oxcert protocol will be driven on both the demand and supply side, notably for:

- a) End users, which will require at least parts of blockchain and NFT functionality
- b) Developers and service providers, which will require faster and more secure development tools to remain competitive in the market.

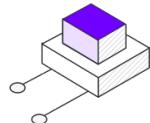
3. Solution

The Oxcert protocol utilizes the blockchain, a distributed ledger technology that was first built to support the Bitcoin cryptocurrency. Blockchain can be best described as a distributed ledger that maintains a list of records called blocks. Each block has a timestamp and is built on top of an already existing block, preventing any data from being altered retroactively.

Blockchain offers a unique way of solving the problem of secure online transactions and double spending. Due to its transparency and distribution of information to many decentralized blockchain nodes, it is nearly impossible to manipulate or duplicate existing data records, making it potentially suitable for recording events, records, identities, certificates, transactions and other documentation.

Oxcert provides a framework with a set of on- and off-chain rules for managing xcerts - standardized and certified non-fungible tokens. Our mission is to equip application developers with a secure blockchain settlement, powerful tools, and community embraced conventions for managing the NFTs.

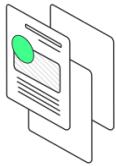
Key characteristics of the Oxcert protocol can be outlined in four larger dimensions:



1. Pluggable settlement for faster development

The Oxcert protocol is a solid pluggable settlement which supports numerous business models. It clears away the low-level blockchain complexity thanks to its solid and flexible infrastructure that ensures interoperability between dapps by default.

Using a plug-and-play framework shortens the development time from months to days. Readily available APIs and SDKs in Python, Ruby, NodeJS and Javascript will allow traditional developers to start building blockchain applications right away.



2. Conventions for data interoperability and standardisation

The Oxcert protocol provides conventions for minting certified non-fungible tokens for unique assets. **With the use of the protocol, proof of an asset can be written into the token directly. These proofs are built following industry-specific conventions and enable data interoperability among various applications.**

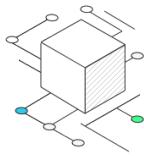
With the help of the community, we are structurally establishing a minimum standard that prevents data-siloing and poor user experience with future implementations.



3. Decentralized verification registry through curated registries

An important piece of the proposed protocol will be authenticating and assuring the validity of issuers building with the Oxcert protocol. Each NFT issued requires a deployed main (token) contract and a mechanism that will attest the validity and the issuers identity.

We are proposing a token curated registry to pursue a decentralized approach in validating the issuers of NFTs. Ultimately, a new issuer would stake a portion of tokens to get listed, while the existing holders will have the ability to approve or challenge the listing.



4. Open source and blockchain agnostic

Oxcert is an open-source and community driven project. Its first implementation is built on Ethereum. However, due to the blockchain-agnostic nature of the protocol, expansion to other blockchains will follow.

The protocol may also include a decentralized governance (DAO) mechanism to allow the community to vote for further improvements. Because the project is open-source, the community will also be able to rely on industry experts for specific conventions in particular verticals. This in turn may positively impact growth of new business models built on top of the protocol.

4. Oxcert Protocol

Oxcert is an open-source permissionless protocol for non-fungible tokens on the blockchain. These tokens are stored in cryptographic wallets and are owned by users. In addition to various common functions for transferring and managing standard non-fungible tokens, the Oxcert protocol provides an additional layer of conventions for creating non-fungible tokens from unique assets. These tokens are called Xcerts and are created through a custom minting process.

Xcerts represent standard non-fungible tokens, which also hold an imprint of an asset. With the Oxcert protocol, we can further validate proof of existence, authenticity and ownership of these assets without third-party involvement.

As opposed to Xcerts, which are non-fungible tokens, the Oxcert protocol also makes use of a fungible ERC-20 utility token, called ZXC. Xcerts are all unique tokens that carry certain information, whereas ZXC tokens are uniform and are used for various utilities described later in the document.

The first implementation of the Oxcert protocol is focusing on the Ethereum blockchain. The Oxcert protocol is designed to be blockchain agnostic, and supports building on other blockchains as well.

In this section, we will provide an overview of basic principles of Oxcert protocol with focus on non-fungibles and decentralization. This is followed by a specification of the Oxcert protocol and framework. The section concludes with description of framework parts: Devkit, decentralized exchange and decentralized miner.

4.1 Overview

The advent of blockchain has redefined technology and set the course for the future of everything. But despite being an amazing technology, it is also very complex in nature. Writing and deploying smart contracts is difficult and can be a very perilous task. This fact prevents many people from adopting blockchain technology and building their decentralized applications on top of it.

Our mission is to equip application developers with a secure blockchain settlement, powerful tools and community embraced conventions for managing non-fungible tokens. Oxcert protocol extends the non-fungible paradigm with an opinionated certification and standardization layer for unique assets, which is based on the Oxcert conventions. This allows for creating certified non-fungible tokens on the blockchain, which also carry an imprint of a unique real-world asset.

The protocol supports a wide range of use cases where non-fungible assets and ownership play a role. Because the data are stored in decentralized blocks, the information can be fully trusted and verified by anyone and anywhere.

The Oxcert protocol is an open-source project so anyone can use the fully functional Oxcert protocol with no limitations. Developers have the ability to manually mint, burn, verify and transfer Xcerts. The protocol uses a publicly accessible network of digital wallets and smart contracts on the blockchain, making it extensible through third-party modules and a variety of dapps.

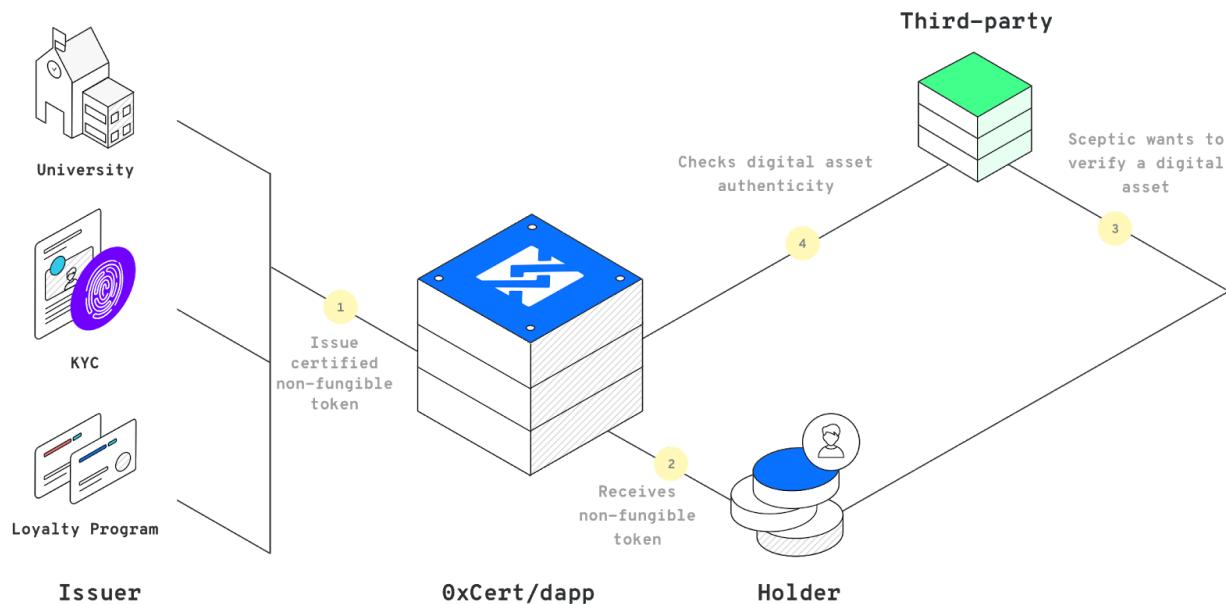


Figure 1 : Oxcert protocol can act as an intermediary between parties.

Oxcert is an opinionated framework and supports numerous business models used by third-party dapps. These applications sit on top of the protocol and can use the protocol tokens as a fuel for their service.

The dapps form a network of public and private services for unique assets and offer higher-level features that simplify and automate the process of creating and managing non-fungible tokens, provide public and private listings, rewarding mechanisms, integration gateways and more.

4.1.1 Fungibility

The most common tokens of today's crypto economy follow Ethereum's ERC-20 specification. These tokens are so-called fungible tokens, because tokens of the same kind can be mutually interchangeable. If we make an analogy with fiat currency, a dollar bill can be exchanged for any other dollar bill, which does not create any difference for the holder.

Recently, another kind of token called non-fungible tokens started getting attention in the crypto community. Though we knew non-fungible tokens before, it actually all started with Crypto Kitties - tradable collectibles, which set the foundation for the now accepted ERC-721 standard. Unlike the ERC-20 identical tokens, the non-fungible tokens are unique and carry data.

The Oxcert protocol goes even further and introduces an Xcert as a standardized and certified non-fungible token based on ERC-721 and Oxcert conventions that carries information about a particular unique asset. This mechanism is unique to the Oxcert protocol and is described in later sections.

ERC-721 has given us an incredibly powerful standard on the Ethereum network - non-fungible tokens. The Oxcert protocol extends this standard and makes it opinionated. This will drastically shorten development times. For example, think about what Rails did for Ruby. Oxcert is dealing with the application layer where developers need to act fast and agile. In order to ensure interoperability among applications in the future, Xcerts follow specific conventions. This will prevent incompatibility on a higher level, which might happen if every NFT issuer deploys their own version of an industry standard. Having this level of standardization built on top of a strong ERC-721 standard prevents high level fragmentation and safeguards long-term sustainability.



Fungible tokens



Non-fungible tokens



Oxcert tokens

4.1.2 Decentralization

The protocol utilizes the blockchain, a distributed ledger technology that was first built to support the Bitcoin cryptocurrency. Blockchain is best described as a distributed ledger that maintains a list of records called blocks. Each block has a timestamp and is built on top of an already existing block, preventing any data from being altered retroactively.

Blockchain offers a unique solution to the problem of secure online transactions. Due to its transparency and distribution of information to many decentralized blockchain nodes, it is nearly impossible to manipulate existing data records, making it potentially suitable for recording events, records, identities, certificates, transactions and other documentation.

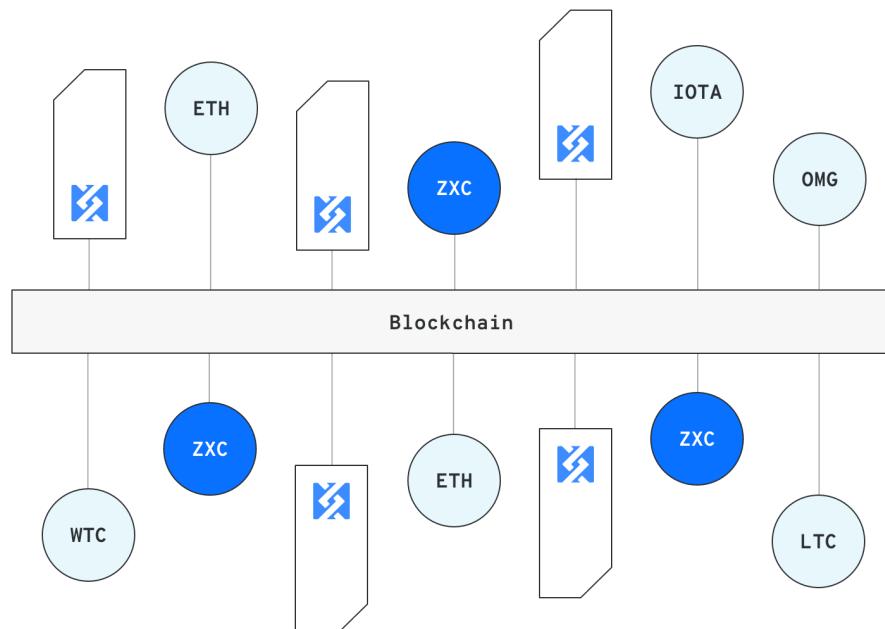


Figure2: Oxcert protocol uses blockchain to enable decentralization

By storing hashed data on the blockchain, individuals, companies and institutions can keep a decentralized record of their asset proofs, while maintaining sensitive data completely private. At the same time, all certified records, their issuers, and owners can be easily authenticated and referenced.

4.2 Specification

Oxcert provides a framework with a set of on-chain and off-chain rules for managing Xcerts and other non-fungible tokens. In addition, the Oxcert protocol is a pluggable settlement with an advanced integration layer for different dapps and relay applications.

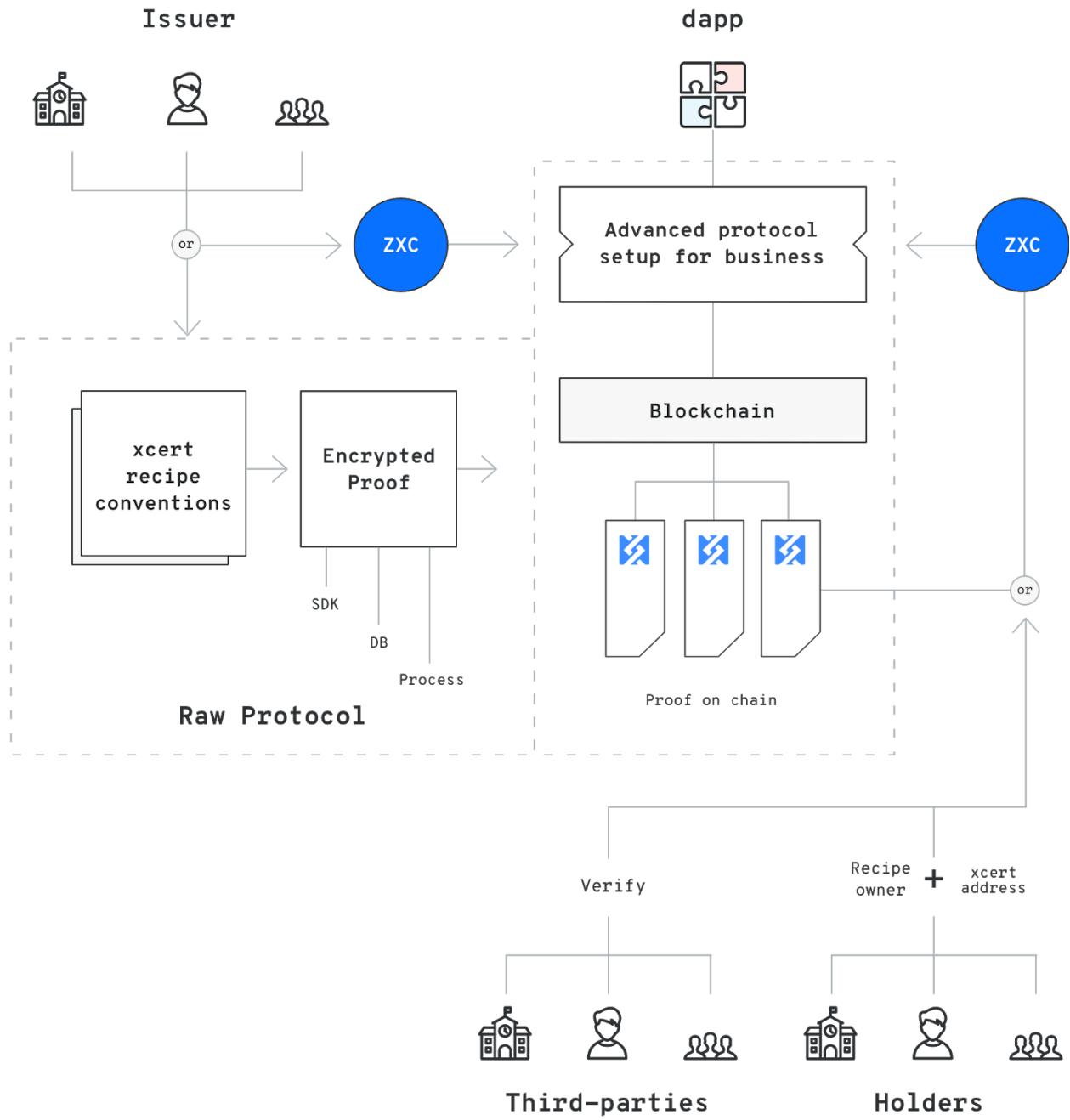


Figure 3: Users can interact with the protocol manually or through higher-level dapps.

4.3. Xcert

Digital assets on the blockchain as non-fungible tokens represent a new paradigm, which enables application developers to build advanced and secure applications for

managing real-world assets. Users hold asset ownership rights in their cryptographic wallets. They are able to exchange tokens between wallets and have control over the tokens they own.

The Oxcert protocol puts unique assets on the blockchain as non-fungible tokens. These tokens are called Xcerts and exist on the blockchain as an item of a specifically designed smart contract.

Xcerts are items of a specifically designed smart contract which implements the non-fungible functionality. An Xcert is identified by an ID, which is unique per Xcert smart contract and is always assigned to a cryptographic wallet. Besides the unique ID, a token can carry proof of a digital asset, arbitrary onchain data and a URI to additional off-chain data.

An Xcert smart contract is an extended non-fungible token smart contract. It follows the Ethereum's ERC-721 specification making it compliant with the non-fungible token standard. Furthermore, an Xcert smart contract holds assets of a particular Oxcert convention. This makes the contract opinionated and forces predictable data.

The process of converting unique asset data into a cryptographic proof is called "certification". Tokens that hold these cryptographic proofs are thus called certified tokens.

4.3 Framework

In general, software frameworks provide a standard way to build and deploy applications. The goal is to simplify the development process and allow programmers to avoid low-level details altogether by providing a working system - a framework. Better known frameworks include Rails (Ruby), Django (Python), Laravel (PHP) and Sails.js (Node.js).

The Oxcert framework consists of multiple parts, which enable application developers to build secure decentralized applications with the support for custom business models. The Oxcert protocol is an opinionated all-in-one framework. It is blockchain agnostic, provides conventions and includes powerful tools for building decentralized non-fungible applications.

In addition to the raw protocol logic, the Oxcert framework includes libraries and a set of smart contracts already installed on the blockchain. The Oxcert framework is a pluggable settlement with an advanced integration layer for different dapps and relay applications.

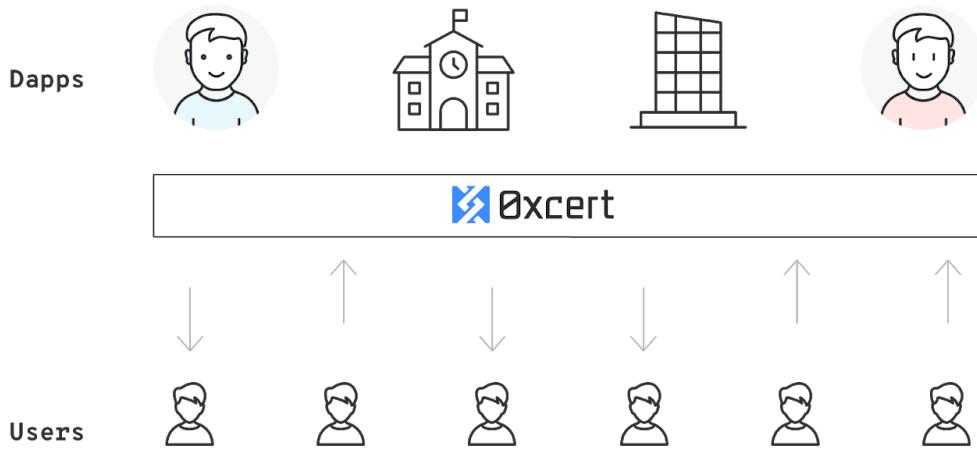


Figure 4: Dapps on top of the Oxcert protocol form a network of interoperable non-fungible services.

4.3.1 Devkit (SDK)

The framework tries to hide away the complex blockchain layer thereby wrapping the Oxcert protocol features into an easy to use SDK. This empowers developers with powerful decentralized tools that they can use as a standard API library.

The Oxcert protocol can therefore be easily integrated into existing systems. Applications don't have to struggle with the low-level blockchain complexity and can immediately start using a solid, secure and flexible non-fungible infrastructure that ensures interoperability between dapps by default.

4.3.2 Decentralized exchange

DEX is one of the key supporting components in the Oxcert framework. DEX represents a set of smart contracts installed on the blockchain which allow for a trustless

exchange of multiple different types of fungible and non-fungible tokens as single atomic operations.

DEX consists of multiple contracts. To make it upgradeable, the smart contracts communicate through proxy smart contracts. This way we can upgrade the core DEX contracts while the data is kept untouched.

A proxy is best explained as a smart contract that allows or rejects access to some key functionality and is controlled by a multisig wallet or a DAO. This way proxies can be trusted since they only allow access to smart contracts that were approved through the DAO process.

4.3.3 Decentralized minter

DXM is another important component of the Oxcert framework. It allows for the trustless minting of Xcert tokens directly to recipients.

The DXM can mint and exchange Xcert tokens for other fungible and non-fungible tokens in a single atomic operation. It also enables an issuer to delegate the mint execution and payment of blockchain fees to the recipient.

DXM consists of multiple contracts. To make it upgradeable, the smart contracts communicate through proxy smart contracts. This way we can upgrade the core DXM contracts while the data is kept untouched.

4.3.4 Continuous integration

A smart contract cannot be changed after it is deployed to the blockchain. Changes can be applied only by deploying a new contract at a new address.

The protocol may include a decentralized governance (DAO) mechanism to allow the community to vote for improvements and possibly fork the protocol into multiple versions. The contracts may use protocol tokens to securely drive a decentralized continuous integration of updates with no disruption, while also protecting all the parties and stakeholders.

5. Use cases

The Oxcert protocol is best described as the underlying technological base layer upon which new use cases can be built. Developers will no longer need to have advanced Solidity skills to create blockchain based solutions for unique assets. The simple to use libraries allow for fast and safe production, which in turn allow for more creativity and flexibility with subsequent solutions.

Below are some examples of use cases where the Oxcert protocol might prove to be useful. It is worth pointing out that this is not a final list of future implementations, but rather a quick overview of potential verticals that can benefit from the non-fungible technology.

5.1 KYC

As the “Know Your Customer” procedure becomes more and more important for ICOs, there is a growing demand for simple, cost-effective and secure solutions. Currently there are many KYC providers on the market, but none of them utilize the powerful of ERC-721 standard for non-fungible tokens.

With the help of the new standard, users can obtain a reusable KYC token, which can be stored and used for every subsequent KYC request. Users simply submit personal data to their KYC provider from which a secure hash is generated. This hashed information can then be sent in the form of an Xcert to their Ethereum wallet. By interacting with a dapp (e.g. through Metamask), ICOs will be able to whitelist wallets that have previously passed the KYC procedure and hold a valid KYC Xcert token.

5.2 Academic credentials

Although technology has helped to improve the education and academic sector immensely, it has not solved major pain points: certification fraud, interoperability and credentials verification remain unsolved questions that in turn delegitimize certification institutions, impair international mobility and incur huge costs for all parties involved. The authentication and verification process can benefit greatly from blockchain technology.

By storing records of achievements, accomplishments, certifications and education degrees on the blockchain, users can keep a decentralized record of their certificates from both academic institutions and professional certifying bodies that can be easily attested.

There are three major parties in the academic credentials space: issuers (universities, MOOCs, ...), holders (students that successfully finished a degree or course) and verifiers (parties that need to verify the authenticity of the academic credential, such as employers or LinkedIn).

With the help of the Xcert protocol, issuers can seamlessly issue academic credentials to the holder's digital wallet. From each academic achievement, a hash string can be created and stored as an Xcert. The non-fungible token that carries the hash string is sent to the holder's wallet and stored as proof for future use. As opposed to previous solutions where the hash string was stored directly on the blockchain as a transaction, users now receive an actual unique digital asset.

Xcerts provide a much more flexible solution than a simple transaction on the blockchain. They can be reissued, have an expiry date and can be easily verified. Just imagine that an error was made when issuing an academic degree, which is then forever stored on the blockchain. Having the flexibility to reissue it allows for a small margin of error.

Certain continuing professional development (CPDs) also need to be renewed and often come with an expiry date. This would be extremely problematic without non-fungible tokens. Having that flexibility where an expiry date can be set to a certificate opens up a whole new space for new verticals and business models.

5.3 Art

Pieces of art are an important part of our cultural tradition; we could even say that it makes us human. Unfortunately, artwork is often the target of forgery, scams and fraud. Cases where entire collections were found to be forged are plentiful and have been a repetitive pattern throughout history.

The art industry is one of the largest unregulated markets, therefore investigating authenticity is strongly advised for buyers prior to a purchase. First, by performing extensive due diligence with the help of independent third parties. Second, with provenance investigation or history of ownership, the details about previous owners can be tracked in documents or other sources.⁴

For the first time in history, the art world has a chance to transfer ownership rights onto a medium, which cannot be altered and falsified. Provenance can easily be traced and viewed for the entire history of each work of art.

The process can be easily achieved with the Oxcert protocol. Data about an asset, e.g. Certificate of Authenticity (COA)⁵, is digitized using a specific convention, stored in cryptographic wallets and owned by their users. The detailed data itself can be accessed only by the owners. However, its existence, authenticity and ownership can, in turn, be examined and validated by the interested public without any third-party involvement.

This opens up a whole new space for artists. Not only provenance and forgery prevention, but also with blockchain technology a whole new concept is possible - fractional ownership.

⁴ <https://medium.com/0xcert/millions-art-fakes-and-blockchain-7a7cb80a52a>

⁵ <http://www.artbusiness.com/certaut.html>

5.4 Collectibles

From sports cards to Crypto Kitties, these are just a few forms of collectibles. When talking about collectible cards in the physical world, many different options come to mind: baseball cards, airplane cards, tradable collectible games (TCG), etc.

Whatever the case may be, serious collectors are interested in vintage cards. Among those, the highest values are attributed to rookie cards, inserts, sets and unopened sets. Other major factors collectors would consider when determining a card's value are its age, origin, condition, scarcity and of course, the featured player. The rarest cards usually come from limited editions and are among the oldest on the market. Values of these cards span from a few dollars to a few million dollars.

In order for a collector to get the most accurate price estimation, several sources of examination and grading are required. Major sports cards grading and autograph authentication authorities, such as Beckett⁶, PSA⁷, and Collectors Universe⁸, now rely their grading business mostly on online demand, while several guides for pricing are issued by Price Guide⁹ and Krause Publications¹⁰, editing Tuff Stuff¹¹ magazine and Sports Collectors Digest¹².

In addition, collectors also need to care of various aspects like the total amount of specific cards on the market, the number of collectors, amount of available ownership data, card provenance and of course the price.

Projects like Crypto Kitties are basically digitizing the very essence of the card collectors hobby, which is the joy of owning something unique and the thrill of comparing or even trading it with others. Even though card editions are issued in many hundreds of (equal) pieces, as soon as they land into the hands of a new owner,, their value cannot be measured as a static feature.

With help from the Oxcert protocol, we can now easily translate real world asset ownership to the blockchain. In the sports cards collecting field, the authenticity could

⁶ <https://www.beckett.com/>

⁷ <https://www.psacard.com/>

⁸ <https://www.collectorsuniverse.com/>

⁹ <https://www.priceguide.cards/en>

¹⁰ <http://www.collect.com/>

¹¹ <http://www.tuffstuff.com/>

¹² <http://www.sportscollectorsdigest.com/>

be traced to the card's manufacturer creating a digital imprint of each card issued. Therefore, when a buyer or collector finds a special insert in the set, they could trace its origin back to the issuer and verify it using the smart contract, without having to rely on third party authentication institution.

When interested in such an asset, collectors could check and verify every single transaction recorded in its data imprint, making it easier for them to evaluate the card before purchase. After acquiring it, they would be able to tokenize it on the blockchain and/or store it in their digital wallet, with ownership and management decisions solely in their hands, giving them the freedom to sell, transfer, burn, or just store it until a good opportunity for trading arises.¹³

¹³ <https://medium.com/0xcert/erc-721-hitting-a-home-run-77d6b4fca33d>

6. Ecosystem

The Oxcert team is determined to continuously bring value to the open-source community. Our mission is to empower developers with powerful tools and useful applications of non-fungible tokens.

In addition to the Oxcert protocol development, the purpose of the Oxcert team, as the core team behind the Oxcert protocol, is to provide a foundation for trustless, certified, non-fungible tokens on the blockchain and to manage and unify the community by connecting individuals and groups working in the area of non-fungibility and to provide resources and support for the related community driven incentives.

The Oxcert ecosystem is made up of various stakeholders, who all have different roles. Due to the nature of the project, a wide spectre of groups will be involved, ranging from research and development, all the way to for-profit companies. In a generalized sense, the Oxcert ecosystem is made up of four major components: Oxcert Labs, Oxcert Protocol Development, Oxcert Protocol Users and the Non-Fungible Alliance.

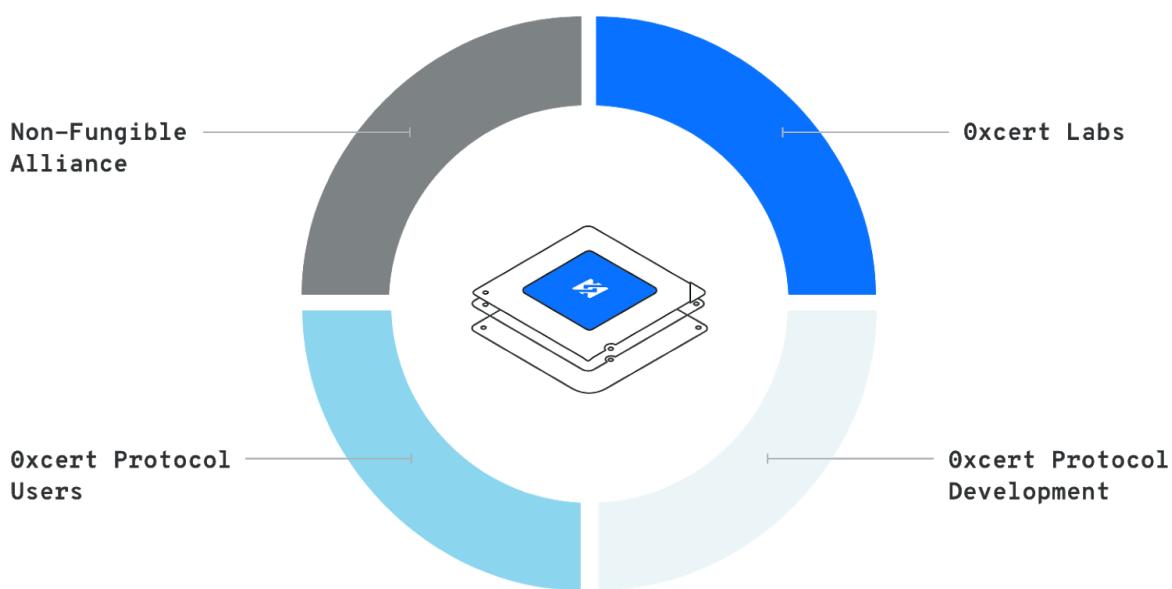


Figure 5: The Oxcert ecosystem

1. Oxcert Labs

We believe that we can only understand developers, the blockchain and decentralization, if the community uses the protocol and builds on top of it. We formed Oxcert Labs as the Oxcert discovery group, which will work on the latest innovations in the space of non-fungibility, decentralization and blockchain technology. Research and development are core pillars that will advance the frontiers of blockchain usability and ultimately drive adoption.

2. Oxcert Protocol Development

Since the Oxcert protocol is an open-source project, we strongly believe that its future development should be community-driven. Down the line, we might implement a full DAO for protocol governance, which would completely empower the developer community.

3. Oxcert Protocol Users

Adoption is key. A great amount of our time, energy and resources will be directed toward ensuring adoption of Oxcert technology. This idea is also completely in line with why we are not just developing a protocol, but rather a full framework with extended tools for fast adoption. The blockchain space in general has to be driven by wider adoption, which cannot happen if each new project has to develop all their smart contracts from scratch. New blockchain projects, as well as, traditional non-blockchain companies will be a key driving force in the future years.

4. Non-Fungible Alliance

As the space of unique digital assets begins to grow, it makes sense to have a collaboration space where all participants can come together. The NFT Alliance is a collaborative hub for building real-life applications with non-fungible tokens technology. It is an association of corporations, services providers and developers that closely collaborate on the implementations of the unique assets on the blockchain.

6.1 Participants

The Oxcert ecosystem consists of a large set of different parties that have come together in order to create and shape the non-fungible space.

Oxcert Ecosystem components	Oxcert Labs	Oxcert Protocol Development	Oxcert Protocol Users	Non-Fungible Alliance
Oxcert Ecosystem participants	Developer community Researchers Partnerships	Developer community Partnerships	Partnerships Developers Projects ICOs Companies	Existing NFT projects and companies

6.2 Promotion & Growth

Each of the four segments described above may have accompanying promotional and growth activities. These range from events and meetups to more comprehensive hackathons and implementation seminars. The Oxcert project is built with large scale adoption in mind, hence growth strategies are also planned.

6.3 Partnerships

The Oxcert protocol is a concept that brings together various actors from different segments and organizations. Establishing mutually beneficial partnerships will be one of the key progress drivers for the organization. Building strategic alliances with complementary organizations, companies and individuals can increase the value of the Oxcert protocol and widen its integrational scope even further.

Oxcert is building partnerships with stakeholders, which are relevant for its success both in the short-run as well as in the long-run. Some of the entities we have already established partnerships with are world leading advisory firms to enable us compliant operations as well as leading institutions promoting certification and the benefits it enables.

Currently we are cooperating with various blockchain organizations and startups, as well as currently in talks to make this cooperation public. Future partnerships will be announced on a rolling basis. We will strive to create meaningful and impactful relations with key players that can further improve the protocol or can assist in various verticals where key players specialize.

PARTNERS



PROJECTS BUILDING ON Oxcert



6.4 Ecosystem Growth Pool

As pointed out, the Oxcert ecosystem is a large structure of various parties and activities. We have dedicated a large amount of the token pool, as well as, funds raised to the growth and promotion of the whole ecosystem. In total, 12% of the entire token supply (Community pool) and 10% of funds raised (Ecosystem) will go towards growth of the Oxcert ecosystem.

7. Token economy

According to William Mougayar, author of "The business blockchain", a token is "a unit of value that an organization creates to self-govern its business model and empower its users to interact with its products while facilitating the distribution and sharing of rewards and benefits to all of its stakeholders." In this section, we present the Oxcert token purpose and characterization, token use cases and their implementation.

7.1 Token purpose and Use cases

The ZXC token is a protocol token and is introduced to align issuing parties with dapps and the community. With the infrastructure built around a system of smart contracts, its primary role is to provide the incentive mechanisms and to support the ecosystem with minimum possible fees.

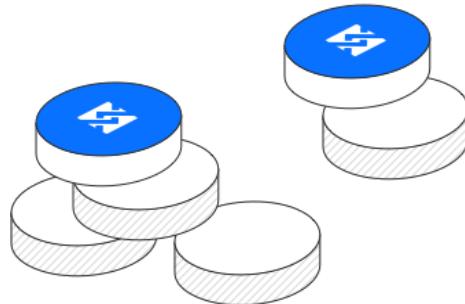


Figure 7: ZXC token is a native utility token of the Oxcert protocol.

The ZXC token is also a part of the extended Oxcert ecosystem that spans beyond the protocol itself. We envision its usage on a few layers, which further decentralize important elements of the corresponding ecosystem.

7.1.1 ZXC within dapps

Dapps developed by Oxcert (e.g. KYC and Academia pilot projects), may use the ZXC token as the basic liquid asset for these dapps to operate on the protocol. Similar to gas on the Ethereum blockchain, the protocol token may play a role in all sorts of protocol and dapp activities. The following use cases are foreseen, but not limited to, within a wide variety of dapps:

- Medium of exchange (payment utility for fees and other costs within dapp ecosystems)

- Staking ZXC tokens within dapps that would require it
- Bidding ZXC tokens for accessing services and/or verification
- Granting access to certain features of the dapp
- Reward and loyalty mechanisms

7.1.2 Decentralized governance

Over time, a decentralized governance may be introduced to further strengthen the Oxcert community. As stated above, Oxcert aims to be a community driven project. Various stakeholders come together to co-create an ecosystem around the protocol itself in an effort to cast an even wider net. In this light, the decentralized governance model may be introduced in the future to drive protocol-level development updates, conventions and update integration.

Token purpose

The protocol token may play a key role when creating a decentralized autonomous organization. The protocol token may not only be used as a rewarding mechanism, but also as a distributed voting mechanism.

7.1.3 Issuer Verification Registry (IVR)

It is becoming apparent that one of the major problems in the future may not be the technological barrier to issuing ownership rights of different unique assets on the blockchain, but rather the authenticity of issuing entities. The underlying objective is to create a self-sustaining curation ecosystem, which could exist without a centralized authority or even the creators themselves.

Currently, the authenticity of issuers is ensured through centralized authorities. For example, higher education institutions have to go through a quality assurance process, which is carried out by an external body to verify if certain standards are met. This is called higher education accreditation that can also be viewed as a curated list or registry of quality educational institutions.

The Oxcert ecosystem may introduce an Issuer Verification Registry (IVR), which would be a form of a token curated registry as proposed by Mike Goldin. This would represent a step toward further decentralization of the whole non-fungible assets space on the blockchain.

As pointed out above, one of the fundamental future problems in the blockchain space may become the authenticity of issuers. This is an issue that touches upon everyone involved in the ecosystem. End-users aspire to have only legitimate issuers and issuers want to be recognized as legitimate entities.

Due to the fact that the Oxcert protocol with its corresponding libraries is completely open source, there is a chance that these tools may be used by entities with ill intent. For example, if a well recognized education provider (e.g. Stanford) wants to issue academic credentials on the blockchain, they can do so using free available tools. Since this education provider maybe a well known and respected institution, there is no objection to them issuing tokenized academic credentials. An issue may arise if a diploma mill poses itself as the same well recognized education provider (e.g. Stanford). It could go ahead and start issuing tokenized academic credentials that may look the same as the originals. This is in fact a scenario that happens in real life and can be detrimental to the educational institution, it can cause a substantial and unnecessary economic cost. End-users and issuers are both affected by this matter. In order to prevent this from happening, a decentralized issuer verification registry maybe introduced.

Token purpose

At the moment, there is no strong incentive that would award fairness and honesty in the blockchain space. A self-sustaining verification registry, which would use staking and rewarding mechanisms based on the Oxcert protocol token may answer many open questions in this relation. The Issuer Verification Registry (IVR) may use the ZXC token to give curation rights that correspond to the relative token weight of token holders. This is a floating ecosystem which has to be decoupled from price fluctuations of other cryptographic currencies or tokens. Only independent supply and demand of ZXC will create a feedback mechanism that ultimately encourages token holders to maintain and curate a list of authentic NFT issuers.

7.2 Implementation

The ZXC tokens are the native utility tokens of the Oxcert protocol. These are fungible tokens and are compliant with the Ethereum's ERC-20 standard.

8. Structure

Oxcert project is and will be structuring its operations and delivery on its goals and roadmap via different elements, which include: corporation structure, governance, token rules, exchanges and voting rules.

8.1 Corporation Structure

Oxcert is currently and at the time of the token sale organized as a company, registered in Slovenia (European Union), the token generated and the funds collected will belong to this company. The Company is set up according to EU laws and regulations.

For the future development and roadmap, the following legal entities will enable the project to deliver on the planned roadmap:

- a) The Oxcert Protocol Foundation: - will manage the Oxcert as a protocol and will use a proportion of tokens for supporting the development of the protocol developers' and projects' community. Expected time to set up the foundation is in late Q3 2018 or Q4 2018.
- b) The Oxcert-for-profit (existing Company): will function as a service provider to the Oxcert foundation (initial development and handover of code) and will be funded by the funds generated by the token sale to develop the Oxcert protocol and initial two pilot cases (Academia and KYC). More information in the roadmap.
- c) For the purpose of better organisation and legal purposes, both entities (a) and (b) can set up additional entities within planned or new jurisdictions.

8.2 Governance

Both Oxcert Protocol Foundation and Oxcert-for-profit will be governed organizationally at three levels:

- a) Board of Directors: The purpose is the overall supervision of the project progress. Consisting of some of the current Oxcert advisors and Management team members, as well as potentially some new members. The number of members is limited to five (5). The Board of Directors will be set up within three

- (3) months after the token sale or, in the case of foundation, at the time of setting up the foundation. Meetings will be scheduled on a quarterly basis.
- b) Advisory Board: Advisory for strategic related decision making. It will consist of some of current Oxcert advisors as well potentially some new members. The number of members is not limited per se. The advisory boards will be set up within three (3) months after the token sale or, in the case of foundation, at the time of setting up the foundation. Meetings will be scheduled on a quarterly basis.
 - c) Management Board: Daily operations and, operational decisions and execution. Consisting of some of current management as well as potentially new members. The number of members is limited to five (5). The management board will be set up within 1 month after token sale. For the foundation, it will be set up at the time of setting up the foundation.

In addition, Oxcert will produce quarterly reports. Quarterly reports will be produced for each of the entities by the Management Board and confirmed by the Board of Directors. It will be publicly available on communications channels and webpage(s). The first report will be published in the first full quarter after the setting up the foundation or after the crowdsale.

We will actively invite interested stakeholders to contribute with their suggestions and comments on our progress and thus engage developers, users, as well as business partners. The suggestions are not going to be binding, however, they will be taken into consideration as valuable input from a wider ecosystem. The team will utilize existing and future communications channels (e.g. Telegram) to interact with the stakeholders as well as hold regular ask me anything (AMA) sessions, if needed, in addition to other interaction opportunities when deemed appropriate. The team is already and will also in the future be actively engaging in conferences and meetups with various stakeholders to exchange opinions and obtain feedback on work and progress.

8.3 Token

The Oxcert token ZXC and the protocol will be managed by the Oxcert Foundation. The number of tokens will be fixed and cannot be additionally minted. Any changes to the token design will be governed by the Oxcert Foundation and its Board of Directors. The token holders will not have voting rights, but the Management Board and Board of Directors can consult with a wider stakeholder ecosystem: developers, partners, token holders as well as other existing and future stakeholders.

8.4 Exchanges

Oxcert will strive to provide as many diverse opportunities for users to purchase or sell tokens. We will actively engage in partnerships with both centralized and decentralized exchanges. When looking for partner exchanges, we will apply the criteria of legal fit as well as acceptable commercial conditions.

The reach out to exchanges will be done prior to the token sale event, however, due to the nature of business practices of exchanges, any announcements can only be made after the token sale has finished. The Oxcert team can announce and talk about the exchanges only after the official confirmation of both sides (this is suggested by legal partners and exchanges).

We must understand the role exchanges play in the project and how the team views exchange partnerships. Exchanges are partners of the project in terms of that they enable users of the protocol to exchange the tokens in order to use and interact with the protocol.

8.5 Voting

Initially, token holders will not have any voting rights. The governance of both entities is in the hands of the Board of Directors and the Management Board for both Oxcert Foundation and Oxcert-for-profit.

In the future, the Oxcert Foundation can introduce voting rights for token holders in a sense of DAO. This decision and the rules are solely in the hands of the Board of Directors of the Oxcert Foundation.

9. The business side of things

Oxcert company is raising funds through token sales to fund the development of the Oxcert protocol and the establishment of the Oxcert Foundation, which will manage the ZXC token and the protocol. However, the Oxcert for profit company also needs a stable business model and a go-to-market strategy for long-term growth and development.

9.1 Business model

As Oxcert-for-profit will have an in-depth knowledge of Oxcert as a protocol, we will arrange our future revenue streams as follows:

- Advisory to startups and corporations: in implementing NFT in their business models and applying Oxcert protocol. Advisory may include, among other services: development, operational processes and legal support. In the event the knowledge or experience is not available in-house, then we will partner with others to provide it. Both advisory fees or equity are options as remuneration.
- Educational programs (training) and professional conferences: for both startups as well as corporations.
- Development of our own dapps: on top of Oxcert protocol and running these as separate business units under the (partial) ownership of Oxcert-for-profit.

9.2 Go-to-market strategy

For-profit advisory and educational programs will be based on acquiring and maintaining relationships with startups and corporates. We will have a dedicated sales unit, which will reach out to new and existing startup and corporate clients and help them address their challenges related to NFT implementations and solutions. Being the creators of Oxcert protocol, we will leverage our brand and our contacts to enable a steady sales funnel flow.

Based on our overall understanding of market opportunities and challenges, we may decide to enter a specific vertical in order to apply NFTs. We will make sure that we do

not enter into conflict of interest situations with our advisory and educational revenue streams. For each of the identified verticals, the for-profit can fund the initial set-up and solution development, however, the vertical dapp will be carved out as a separate business unit with its management, operations and development team.

9.3 Competitive landscape

The non-fungible token space is just beginning. ERC-721 has shown us that the immense capabilities of blockchain lie in front of us and have not really even been tapped into. Placing unique assets on the blockchain will spur a whole new plethora of ideas, projects and companies -- things previously impossible and unimaginable.

Oxcert is dealing primarily with the application layer of things: giving developers tools to more effectively build applications on top of this new standard. Developers can build dapps on top of it, while end users can use their solutions to prove authenticity, authorship or ownership of their assets (such as collectibles, university degrees, identity/KYC, in-game items or a house), making them secure and available while they are certified on the blockchain.

At the moment of writing this white paper, we were unable to identify a competitor that would be focusing on developing a unifying developer framework with conventions on top of ERC-721. However, there are competitors in various niches where the Oxcert protocol and higher-level dapps built with the Oxcert protocol can be used. These span from art, collectibles, education credentials, identity and more, all the way to real estate and car ownership.

Oxcert protocol is a building block for everyone to use. In that light, we see most of the companies that on first glance may appear as competitors, rather as potential partners and protocol adopters. Using unique digital assets is in many cases a superior solution, hence we will see a wider adoption in the future. Oxcert is merely going to provide the underlying infrastructure for all these projects to use.

The purpose of the Oxcert organization, as the core team behind the Oxcert protocol, is to provide a foundation for trustless, certified, non-fungible tokens on the blockchain and unify the community to the fullest. We intend to bring value to the open-source community engaged with the Oxcert protocol, to connect individuals and groups

working in the area of non-fungibility or certification and to provide resources and support for the related community-driven incentives.

10. Timeline

The inception of Oxcert was in 2017. After the first MVPs were created and tested, a more rigorous and detailed development roadmap was laid out. Although we are on track, there is still a long way to go. Below is a detailed roadmap, which is followed by milestones¹⁴.

10.1 Development roadmap

Q3 2017 ✓	- First proposal of the blockchain certification technology - First MVP for deployment and verification of certificates
Q4 2017 ✓	- Shift towards open-source - Explore wider adoption and a protocol approach - Xcert smart contract draft - Technical paper draft
Q1 2018 ✓	- Pivot towards ERC-721 for assets on the blockchain - Proof of protocol concept - DEX draft implementation
Q2 2018 ✓	- Joining forces with ERC-721 standard lead author, William Entriken - ERC-721 complete implementation with bounty - Xcert complete implementation with bounty - DEX alpha implementation - Xcert minter alpha implementation - Oxcert scanner alpha dapp - Technical paper 1.0
Q3 2018	- Whitelist/KYC certificate dapp (another PoC for crowdsale) - Crowdsale PoC certificates used - Protocol draft on Ethereum mainnet (limited) - DEX alpha dapp
Q4 2018	- Protocol 1.0 (Ethereum) - Odin - Framework 1.0 (protocol features, application layer) - DEX 1.0 dapp - Minter 1.0 dapp
2019 Q1+Q2	- Protocol 2.0 (Ethereum, second chain) - Aragorn - Framework 2.0 (notification system, community requested features)
2019 Q3+Q4	- Pilot dapp for selected vertical 1 - Pilot dapp for selected vertical 2 - Curated registry

¹⁴ The timeline and milestones may be susceptible to change due to unforeseen events, complications and interruptions.

10.2 Milestones

	MVP	Alpha	Beta	Odin (protocol v1)	Aragorn (protocol v2)
Protocol					
Xcert	X (proof of protocol concept)	x	x	x	
Conventions		x	x	x	x
Framework					
Devkit (SDK)		x	x	x	x
DEX implementation		x		x	x
DXM implementation	x	x		x	
Oxcert Labs					
ERC-721 implementation	x	x			
Scanner dapp	x	x	x		
Identity dapp			x		
DEX dapp				x	x
DXM dapp				x	x
IVR				x	x

11. Token distribution event

The Oxcert Token Distribution Event will take place during the months of June and July in 2018. In this section, we present the information about the Oxcert token, crowdsale, token distribution and funds allocation.

11.1 Token and crowdsale info

Token name: ZXC (ERC-20)

Price of token: 0.0001 ETH, 1 ETH = 10,000 ZXC

Max presale bonus: 20%

Hardcap: ~20,000 ETH (estimated based on bonus distribution)

Softcap: 5,000 ETH

Token Supply: 500,000,000 (fixed, no future minting)

Circulating supply: 250,000,000

Percentage of tokens going to contributors in all token sale stages: 50%, 250 mio ZXC tokens.

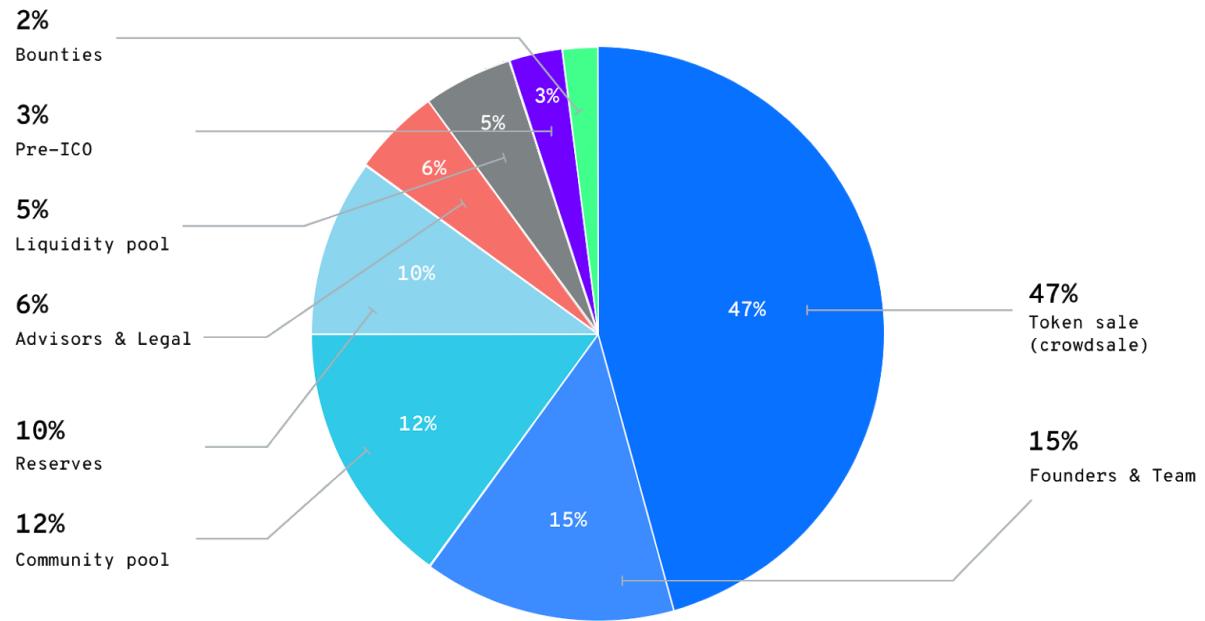
Oxcert token sale consists of four stages: Pre-ICO (early buyers), Private Presale (larger buyers), Public Presale (10% discount) and a Public Crowdsale. Details about each stage can be found in the table below. In case the allocated number of tokens for each stage is not sold in the allocated stage, it is rolled to the next stage. If there are tokens planned for sale left after the final stage, they will be burnt.

Stage	Date	Tokens Allocation	Max bonus	% Total tokens
Pre-ICO	Closed	15,000,000	40%	3%
Private Presale Min. participation 200 ETH	Until June 20th 2018	120,000,000	15-20%	24%
Public Presale Min. participation 1 ETH	July 2 - July 4, 2018	71,157,402	10%	14%
Crowdsale	July 4 - July 18, 2018	43,842,597	5% (first 24 hrs)	9%

11.2 Token distribution

The Oxcert token distribution is found in the table below. 50% of all tokens are reserved to be sold in different stages of the crowdsale. The remaining is reserved for team, current and future stakeholders and future reserves.

Token distribution	% total tokens	Purpose
Pre-ICO	3.00%	Token sale
Token sale (crowdsale)	47.00%	Token sale
Founders and team	15.00%	Incentives alignment
Advisors (& legal)	6.00%	Incentives alignment
Liquidity pool	5.00%	Supporting token liquidity
Community pool	12.00%	Supporting development of protocol community, allocated to Foundation
Bounties	2.00%	Bug bounties, allocated to Foundation
Reserves	10.00%	Future development fund



Distributed tokens differ in their lock-up period. The Founders, Team and Advisors have tokens locked-up in different periods. The buyers do not have tokens locked-up.

Lock-Up for Buyers: None

Lock-up for Founders: Locked for 6 months then 12.5% and 12.5% every three months.

Lock-up for Team: 20% released at ICO and 15% every three months after ICO.

Lock-up for Advisors: 20% at ICO, 40% in three months, 40% in six months

Lock-up for Reserves: 2 years

11.3 Distribution schedule

The distribution of the tokens will be done via the the crowdsale smart contract, with the following design:

- a) The Pre-ICO and Private Presale buyers will receive their purchased ZXC tokens to the ETH wallet address they have provided to Oxcert.

- b) The Public Presale and the Crowdsale buyers will receive their purchased ZXC tokens to the originating ETH wallet address from where the ETH funds to purchase ZXC token have been sent.
- c) All token buyers will receive their ZXC token the latest 7 days after the final date of the public crowdsale.
- d) The unsold tokens allocated to the token sale will be burnt within 7 days after the final end date of the public crowdsale.
- e) The remaining 50% of tokens (not allocated to token sale) will be distributed to their respective ETH wallet addresses within 7 days after the final end date of the public crowdsale.

All the distributed tokens will be locked up until 7 days after the final end date of the public crowdsale. Once the tokens are unlocked, the buyers can move them. This timeline is based on Ethereum network functioning normally and provided there are no lags.

11.4 Participation in the token sale

The participation in the token sale event will require buyers to:

- a) Have ETH (Ethereum) cryptocurrency
- b) Have an ETH wallet address
- c) Do a KYC required for the purchasing of the token
- d) Be a citizen of countries that are eligible for their citizens to participate in the token sale (more details in the Token Sale Terms and Conditions).

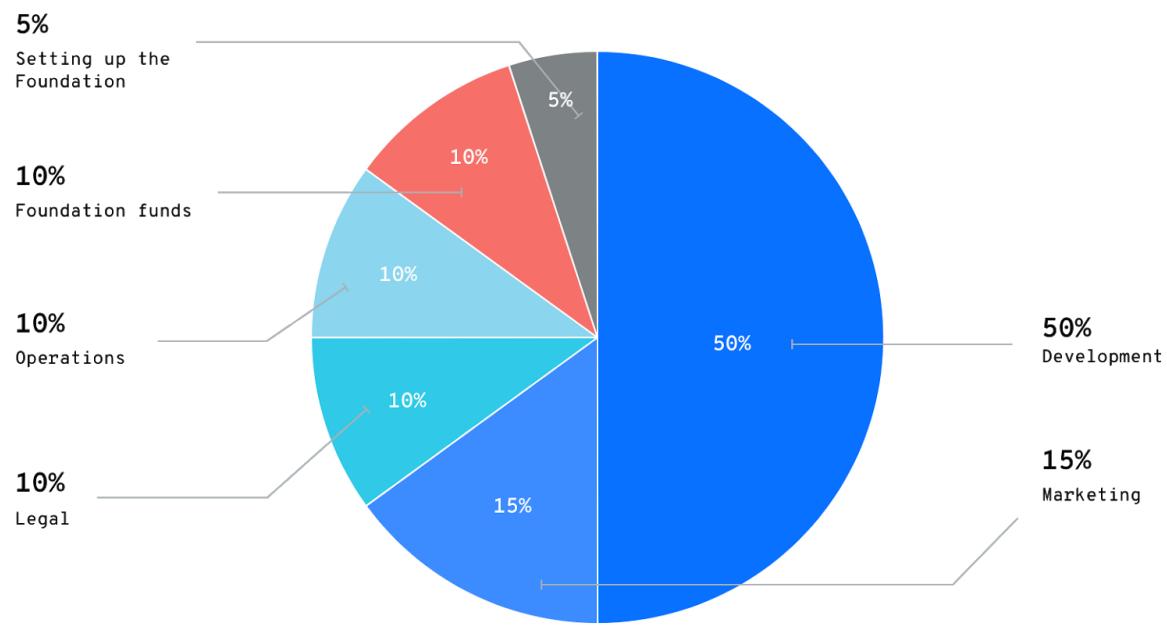
In addition to conditions above:

- e) In order to participate in the private presale there is a minimum of 200 ETH of purchase amount
- f) Only public presale whitelisted buyers can participate in the Public Presale. The public presale whitelist will be closed before the beginning of the public presale. The information when the whitelisting for public presale will be closed will be published on the Oxcert website. In the public presale 1 ETH is the minimum purchase amount
- g) During the public crowdsale the token sale will be open to all (respecting condition a) to d)). There will be no minimum purchase amount in the public crowdsale

11.5 Funds allocation

The funds collected in the token sale will be used with the following allocation:

Funds allocation	% of all funds
Development	50.00%
Marketing	15.00%
Legal	10.00%
Operations	10.00%
Setting up the Foundation	5.00%
Foundation funds	10.00%



The last two allocation areas are aimed at setting up the Oxcert foundation as a separate legal entity as well as transferring a part of funds to the Foundation to enable immediate foundation operations.

12. Team

The Oxcert team is a fully dedicated group of highly skilled individuals. At the moment, the team consists of 18 members that operate in all crucial areas to support both the long-term strategy as well as the short-term execution. In contrast to many other projects and companies, the vast majority of resources are in-house.

12.1 Team

Operations

Jure Zih, CEO

Mitja Pirc, COO

Anja Pukl, CFO

Development team

Kristijan Sedlak, CTO

Tadej Vengust, Solidity developer

David Ličen, Lead front-end developer

Luka Kacil, Senior blockchain developer

Leon Panjtar, Full-stack developer

Gregor Ažbe, Junior front-end developer

Growth team

Jure Jakomin, Growth hacker

Matic Užmah, Customer success

Petra Kosi, Social media

Petra Mišić, Public Relations

Content and design team

Nick Stein, Head of content

Lenka Tušar, Content creation

Romina Kavčič, Design Strategy

Community management team

Urban Osvald, Head of community

Tomaž Železnikar, Community manager

Kim Fairley, Community & Partnership manager

11.2 Advisors

William Entriken, ERC-721 main author

www.linkedin.com/in/fulldecent/, <https://github.com/fulldecent>, <https://phor.net/>

William Entriken is the lead author of ERC-721. He is an active computer science researcher who has contributed tremendous amounts of open source projects to the community. In his non-blockchain life, Will is the General manager of Pacific Medical Training, a company that creates interactive medical training courses.

Moe Levin, Blockchain pioneer

www.linkedin.com/in/moelevin/

Moe Levin is a blockchain evangelist and pioneer. He is the CEO of Keynote, a company focusing on creating blockchain events all over the world. Moe is also an Executive committee member of the Global Blockchain Council and the founder of the North American Bitcoin Conference.

Mark Pui, MW Partners advisor

<https://www.linkedin.com/in/mark-pui-b9ab3046/>

Mark Pui has up to recently been the Executive Director at PwC in Kuala Lumpur, Malaysia. He draws his skills and experience from a long and successful career in management consulting, corporate finance, consulting and advising. Mark is also a seed investor in early stage companies like Ox, Bancor, EOS, QTUM, Tezos and many others. Within the blockchain space, his focus lies in blockchain interoperability, tokenization of financial and non-financial assets, privacy protocols, stablecoins and Enterprise-oriented use cases.

Dr. Draško Veselinović, multi CEO experience

www.linkedin.com/in/draskoveselinovic/

Assoc. Prof. Dr. Draško Veselinović is the President and CEO of SEBRA - Slovenian Business and Research Association. In his career, he held many top executive positions and has more notably co-founded the Yugoslavian Stock Exchange as well as the

Slovenian Stock Exchange. To this day, he remains a well respected authority in the field of business, finance and education.

Dušica Lukač: Fintech expert, C-level

www.linkedin.com/in/dusicalukac/

Dušica is a seasoned financial expert and Founder of Stizzbuzz, a boutique investment and consultancy company with focus on Startups and Tech companies. She has also held numerous top level executive positions in and co-founded many successful business.

Simon Belak, tech entrepreneur

www.linkedin.com/in/simonbelak/

Simon is a tech entrepreneur, highly proficient in data analytics and currently working as a Mad scientist with Metabase. He is able to juxtapoz philosophical views and concepts with advanced science and technology - a unique perspective rarely found.

Dr. Daithí Ó Murchú, President RCEEDAO Ireland

<https://www.linkedin.com/in/dr-draith%C3%AD-%C3%B3-murch%C3%BA-57228624/>

Dr. Daithí Ó Murchú is the President of RCEEDAO Ireland and is under EU parliament and Commission Appointment. He is also the founding member at the International Fellow Academy of Ubiquitous Communication Educators International. He has vast experience from the education segment as well as in technology, innovation, business and management.

Disclaimer

“This paper is for information purposes only and no part of it is intended to create legal relations between a recipient of this paper or to be legally binding or enforceable by such recipient against the company that owns the project. An updated version of this paper may be published on a date to be determined and announced by the company that owns the project in course. Company that owns the project makes no warranties or representations as to the successful development or implementation of such technologies and innovations, or achievement of any other activities noted in the paper, and disclaims any warranties implied by law or otherwise, to the extent permitted by law.”

RISK FACTORS

THIS SECTION ON RISK FACTORS IS NOT AND DOES NOT PURPORT TO BE A COMPLETE ENUMERATION OR EXPLANATION OF THE RISKS INVOLVED WITH THE PURCHASE OF ZXC Token. THERE MAY BE ADDITIONAL MATERIAL RISKS THAT THE DIRECTORS DO NOT CURRENTLY CONSIDER TO BE MATERIAL OR OF WHICH THE DIRECTORS ARE NOT AWARE. THE FOLLOWING THEREFORE HIGHLIGHTS CERTAIN RISKS TO WHICH THE COMPANY IS SUBJECT TO AND WHICH THE COMPANY WISHES TO ENCOURAGE PURCHASER TO DISCUSS WITH THEIR OWN PROFESSIONAL ADVISORS.

Prospective ZXC Token purchasers should conduct such independent investigation and analysis regarding this Company, the ZXC Token and all other relevant market and economic factors as they deem appropriate to fully evaluate the merits and risk of their purchase.

The Company and its Directors disclaim any responsibility to advise purchasers of ZXC Token of the risk and considerations associated with the purchase of ZXC Token as they exist at the date hereof or from time to time hereinafter.

Each prospective purchaser of any ZXC Token must determine, based on his/her own independent review and such professional advice (including, without limitation, tax, accounting, credit, legal and regulatory advice) as it deems appropriate, that the purchase of ZXC Token is appropriate and suitable for it, notwithstanding the clear and substantial risks inherent with the purchase of ZXC Token.

You should consult with your own legal, regulatory, tax, business, investment, financial and accounting professional advisors to the extent that you deem it necessary, and make your own decisions including decisions regarding the suitability of this purchase based upon your own judgement and upon advice from such professional advisors as you deem necessary and not upon any view expressed by any party mentioned in this Whitepaper.

The purchaser of a ZXC Token should be capable of evaluating the merits and risks of such a purchase and should have sufficient resources to be able to bear any losses (which may be equal to the whole

purchased amount) that may result from such a purchase. Prospective purchasers of ZXC Token should be aware that the value of ZXC Token may go down as well as up and that they may not be able realise their purchase amount on the secondary market (if there is any).

Forward looking statements

Certain statements in this Whitepaper constitute “forward looking statements” that are used on the beliefs of the Directors and reflect their current expectations. When used in this Whitepaper or in any of the Company’s material, the words “estimate”, “project”, “believe”, “anticipate”, “intend”, “expect”, “plan”, “predict”, “may”, “might”, “could”, “should”, “would”, “will”, the negative of these words or such other variations thereon or comparable terminology are intended to identify forward-looking statements. Such statements reflect the views of the Directors at the time the statements are made with respect to future events based on information available at that time, and are subject to risks and uncertainties that could cause actual results to differ materially from those contemplated in those forward-looking statements. The Directors assume no obligation to update or revise these statements to reflect current information, events, or circumstances, including changes in any risks or uncertainties that may impact them.

Management Risk

If any of the directors or officers of the Company cease to participate in the operation of the Company, the operations, objectives, and activities of the Company may be adversely affected.

Liquidity of ZXC Token

As of the date of this Whitepaper, there is no active secondary market for the ZXC Token. Whilst the Directors hope that the success of the Company will lead to a secondary market developing, there is no guarantee or assurance that a public market will ever develop. There is often no assurance that a purchaser of the ZXC Token will be able to sell or dispose of the ZXC Token.

Changes in Applicable Law and Regulation

The Directors believe that it is possible that emergency intervention by certain Governments may take place in the future in respect of ICOs. Such intervention may be implemented on an “emergency” basis, subjecting market participants without notice to a set of regulations which in some cases may be unclear in scope and in application.

Should any relevant laws or regulations change, the legal requirements to which the Company and the ZXC Token may be subject could differ materially from current requirements. No assurance can be given that future legislation, administrative rulings or court decisions will not adversely affect the Company and the ZXC Token.

The Company may be subject to a number of unusual risks, including contradictory legislation, incomplete, unclear and changing laws, ignorance or breaches of regulations on the part of other market participants, lack of established or effective avenues for legal redress, lack of standard practices and confidentiality customs characteristic of developed markets and lack of enforcement of existing regulations.

Early Stage Companies

The Company is a start-up and has no operating history against which purchasers of the ZXC Token may consider the appropriateness of purchasing the ZXC Token.

Many risks and uncertainties affect start-up and early stage companies, which often have very limited operating history, profits or cash flow. There can be no assurance of the success of such enterprises. Their potential must be considered in light of the problems, expenses, difficulties, complications and delays frequently encountered in connection with new or developing businesses, including technology risks, unproven business models, untested plans, uncertain market acceptance, competition and lack of revenues and financing.

The technological fields and markets that many start-up and early stage companies address have undergone and are expected to continue to undergo rapid and significant change. Rapid technological developments may result in the technology of companies becoming obsolete, uneconomical or uncompetitive before any commercial success or financial return can be achieved. Numerous other risks may affect developing companies and ventures, including risks that products or services will be found to be ineffective, unreliable, unsafe or uncompetitive and risks that such companies' technologies, products or service will not achieve market acceptance or penetration. Market acceptance of new products, services or technologies depends on many factors and uncertainties and cannot be assured.

Start-up and early stage companies may compete with entities that have established businesses, relationships and positions in the market and that have much more substantial financial, business, technological, marketing and distribution assets, operations and resources. There can be no assurance that any developing company will be able to compete successfully with more established companies.

These companies may be overly dependent on the vision, skill and leadership of a single or limited number of executives. In a start-up business, the loss or disability of a key person(s) can result in significant financial hardship, in some cases the failure of the company.

Any projections, forecasts, plans or other forward-looking statements are subject to numerous risks, uncertainties, changing circumstances and other factors that could cause actual results, performance, plans, prospects, operations and opportunities to differ materially from any forward-looking statements, including competition, inability to identify and do business with appropriate customers, existing and future law and regulations, liabilities under the securities laws, inability to hire, retain or qualify sufficient management and staff, general economic conditions, rapid technological change, cost overruns, delays in bringing products or services to market, marketing failures, difficulty in penetrating markets, delays or failures in developing anticipated capabilities, products or services, failure to obtain necessary regulatory approvals, insufficient funding, lack of availability of capital, rates of economic growth, levels of consumer and business spending, conditions in the technology and financial industries, dependence on strategic partners and business relationships, unproven business models, adverse developments affecting customers and end-users, fluctuations in securities markets and valuations, limited marketing, expansion risks, losses and costs, uncertain revenues and profitability, conditions in particular industries, accounting problems, costs, delays and liabilities arising from legal proceedings, failure to obtain and maintain intellectual property or proprietary rights and management failures.

Banking and custody arrangements

The Company's cash will be held by a bank. The Company acknowledges that any such deposits are not guaranteed by the bank and are exposed to losses incurred in the event of the insolvency or failure of the bank. The Company will take credit risk against any party which is holding its cash. The Company will therefore rank as a general unsecured creditor in the event of the insolvency or failure of the bank with which deposits or instruments have been placed.

Regulatory Supervision

The Company and the ZXC Token are not regulated by the EU or Slovenian Financial Services Commission or any other regulatory or supervisory authority. The EU or Slovenian Financial Services Commission does not vouch for the financial soundness of the Company, the ZXC Token or for the correctness of any statements made, or opinions expressed with regards to it.

Cybersecurity

Cybersecurity threats are present within the realms of cryptocurrencies. There is a risk of loss of funds, including a total loss, should an unauthorised intrusion or theft occur.

Whilst the Company has considered its cybersecurity, risks related to software weakness, human error, external attacks and others, continue to exist and pose a material risk to the Company and the value of the ZXC Token.

Advances in cryptography, or technical advances such as the development of quantum computers, may present risks for crypto-currencies, which could result in the theft or loss of ZXC.

Hackers or other malicious or criminal groups or organizations may attempt to interfere with the Token Sale, the ecosystem or the availability of ZXC in several ways including, but not limited to, denial of service attacks, Sybil attacks, mystification, phishing, attacks, smurfing, malware attacks, or consensus based attacks.

Ethereum Network

The ZXC Token is a part of the Ethereum network. If problems related to the Ethereum network normal functionality arise, this may affect the ZXC Token functionality and may adversely affect the Company and the value of the ZXC Token. Therefore, any malfunction, unplanned function or unexpected operation of the Ethereum protocol may cause the ZXC ecosystem or ZXC to malfunction or operate in a way that is not expected. Ether, the native Ethereum Protocol account unit may itself lose value in a similar way to ZXC, and also in other ways. For more information on the Ethereum protocol, see <http://www.ethereum.org>. Any error in the smart contract may lead to the lose.

As with other decentralized cryptographic tokens and crypto-currencies, the Ethereum blockchain used for the ecosystem is vulnerable to mining attacks, including but not limited to, dual-expense attacks, powerful mining attacks, selfish mining attacks, and critical competition attacks. Any successful attack poses a risk to the software and the expected performance and sequencing of Ethereum contract calculations. Despite the best efforts of the team, the risk of known or new mining attacks exists.

Crypto-currencies and cryptographic tokens are a cutting-edge, untested technology. In addition to the risks stipulated above, there are other risks that the Oxcert team cannot predict. Risks may also occur as unanticipated combinations or as changes in the risks stipulated herein.

THE FOREGOING RISK FACTORS DO NOT PURPORT TO BE A COMPLETE EXPLANATION OF THE RISKS INVOLVED WITH THE COMPANY AND THE ZXC Token.

The Ripple Protocol Consensus Algorithm

David Schwartz
david@ripple.com

Noah Youngs
nyoungs@nyu.edu

Arthur Britto
arthur@ripple.com

This paper does not reflect the current state of the ledger consensus protocol or its analysis. We will continue hosting this draft for historical interest, but it SHOULD NOT be used as a reference. For an updated analysis and presentation of the consensus protocol, please refer to arXiv:1802.07242 (<https://arxiv.org/abs/1802.07242>), released 20 February 2018.

Abstract

While several consensus algorithms exist for the Byzantine Generals Problem, specifically as it pertains to distributed payment systems, many suffer from high latency induced by the requirement that all nodes within the network communicate synchronously. In this work, we present a novel consensus algorithm that circumvents this requirement by utilizing collectively-trusted subnetworks within the larger network. We show that the “trust” required of these subnetworks is in fact minimal and can be further reduced with principled choice of the member nodes. In addition, we show that minimal connectivity is required to maintain agreement throughout the whole network. The result is a low-latency consensus algorithm which still maintains robustness in the face of Byzantine failures. We present this algorithm in its embodiment in the Ripple Protocol.

Contents

1	Introduction	1
2	Definitions, Formalization and Previous Work	2
2.1	Ripple Protocol Components	2
2.2	Formalization	3
2.3	Existing Consensus Algorithms	3
2.4	Formal Consensus Goals	3
3	Ripple Consensus Algorithm	4
3.1	Definition	4
3.2	Correctness	4
3.3	Agreement	5
3.4	Utility	5
	Convergence • Heuristics and Procedures	
4	Simulation Code	7
5	Discussion	7
6	Acknowledgments	8
	References	8

1. Introduction

Interest and research in distributed consensus systems has increased markedly in recent years, with a central focus being on distributed payment networks. Such networks allow for fast, low-cost transactions which are not controlled by a centralized source. While the economic benefits and drawbacks of such a system are worthy of much research in and of themselves, this work focuses on some of the technical challenges that all distributed payment systems must face. While these problems are varied, we group them into three main categories: correctness, agreement, and utility.

By correctness, we mean that it is necessary for a distributed system to be able to discern the difference between a correct and fraudulent transaction. In traditional fiduciary settings, this is done through trust between institutions and cryptographic signatures that guarantee a transaction is indeed coming from the institution that it claims to be coming from. In distributed systems, however, there is no such trust, as the identity of any and all members in the network may not even be known. Therefore, alternative methods for correctness must be

utilized.

Agreement refers to the problem of maintaining a single global truth in the face of a decentralized accounting system. While similar to the correctness problem, the difference lies in the fact that while a malicious user of the network may be unable to create a fraudulent transaction (defying correctness), it may be able to create multiple correct transactions that are somehow unaware of each other, and thus combine to create a fraudulent act. For example, a malicious user may make two simultaneous purchases, with only enough funds in their account to cover each purchase individually, but not both together. Thus each transaction by itself is correct, but if executed simultaneously in such a way that the distributed network as a whole is unaware of both, a clear problem arises, commonly referred to as the “Double-Spend Problem” [1]. Thus the agreement problem can be summarized as the requirement that only one set of globally recognized transactions exist in the network.

Utility is a slightly more abstract problem, which we define generally as the “usefulness” of a distributed payment system, but which in practice most often simplifies to the latency of the system. A distributed system that is both correct and in agreement but which requires one year to process a transaction, for example, is obviously an inviable payment system. Additional aspects of utility may include the level of computing power required to participate in the correctness and agreement processes or the technical proficiency required of an end user to avoid being defrauded in the network.

Many of these issues have been explored long before the advent of modern distributed computer systems, via a problem known as the “Byzantine Generals Problem” [2]. In this problem, a group of generals each control a portion of an army and must coordinate an attack by sending messengers to each other. Because the generals are in unfamiliar and hostile territory, messengers may fail to reach their destination (just as nodes in a distributed network may fail, or send corrupted data instead of the intended message). An additional aspect of the problem is that some of the generals may be traitors, either individually, or conspiring together, and so messages may arrive which are intended to create a false plan that is doomed to failure for the loyal generals (just as malicious members of a distributed system may attempt to convince the system to accept fraudulent transactions, or multiple versions of the same truthful transaction that would result in a double-spend). Thus

a distributed payment system must be robust both in the face of standard failures, and so-called “Byzantine” failures, which may be coordinated and originate from multiple sources in the network.

In this work, we analyze one particular implementation of a distributed payment system: the Ripple Protocol. We focus on the algorithms utilized to achieve the above goals of correctness, agreement, and utility, and show that all are met (within necessary and predetermined tolerance thresholds, which are well-understood). In addition, we provide code that simulates the consensus process with parameterizable network size, number of malicious users, and message-sending latencies.

2. Definitions, Formalization and Previous Work

We begin by defining the components of the Ripple Protocol. In order to prove correctness, agreement, and utility properties, we first formalize those properties into axioms. These properties, when grouped together, form the notion of *consensus*: the state in which nodes in the network reach correct agreement. We then highlight some previous results relating to consensus algorithms, and finally state the goals of consensus for the Ripple Protocol within our formalization framework.

2.1 Ripple Protocol Components

We begin our description of the ripple network by defining the following terms:

- **Server:** A server is any entity running the Ripple Server software (as opposed to the Ripple Client software which only lets a user send and receive funds), which participates in the consensus process.
- **Ledger:** The ledger is a record of the amount of currency in each user’s account and represents the “ground truth” of the network. The ledger is repeatedly updated with transactions that successfully pass through the consensus process.
- **Last-Closed Ledger:** The last-closed ledger is the most recent ledger that has been ratified by the consensus process and thus represents the current state of the network.
- **Open Ledger:** The open ledger is the current operating status of a node (each node maintains its own open ledger). Transactions initiated by end users of a given server are applied to the open

ledger of that server, but transactions are not considered final until they have passed through the consensus process, at which point the open ledger becomes the last-closed ledger.

- **Unique Node List (UNL):** Each server, s , maintains a unique node list, which is a set of other servers that s queries when determining consensus. Only the votes of the other members of the UNL of s are considered when determining consensus (as opposed to every node on the network). Thus the UNL represents a subset of the network which when taken collectively, is “trusted” by s to not collude in an attempt to defraud the network. Note that this definition of “trust” does not require that each individual member of the UNL be trusted (see section 3.2).
- **Proposer:** Any server can broadcast transactions to be included in the consensus process, and every server attempts to include every valid transaction when a new consensus round starts. During the consensus process, however, only proposals from servers on the UNL of a server s are considered by s .

2.2 Formalization

We use the term *nonfaulty* to refer to nodes in the network that behave honestly and without error. Conversely, a *faulty* node is one which experiences errors which may be honest (due to data corruption, implementation errors, etc.), or malicious (Byzantine errors). We reduce the notion of validating a transaction to a simple binary decision problem: each node must decide from the information it has been given on the value 0 or 1.

As in Attiya, Dolev, and Gill, 1984 [3], we define consensus according to the following three axioms:

1. **(C1):** Every nonfaulty node makes a decision in finite time
2. **(C2):** All nonfaulty nodes reach the same decision value
3. **(C3):** 0 and 1 are both possible values for all non-faulty nodes. (This removes the trivial solution in which all nodes decide 0 or 1 regardless of the information they have been presented).

2.3 Existing Consensus Algorithms

There has been much research done on algorithms that achieve consensus in the face of Byzantine errors. This

previous work has included extensions to cases where all participants in the network are not known ahead of time, where the messages are sent asynchronously (there is no bound on the amount of time an individual node will take to reach a decision), and where there is a delineation between the notion of strong and weak consensus.

One pertinent result of previous work on consensus algorithms is that of Fischer, Lynch, and Patterson, 1985 [4], which proves that in the asynchronous case, non-termination is always a possibility for a consensus algorithm, even with just one faulty process. This introduces the necessity for time-based heuristics, to ensure convergence (or at least repeated iterations of non-convergence). We shall describe these heuristics for the Ripple Protocol in section 3.

The strength of a consensus algorithm is usually measured in terms of the fraction of faulty processes it can tolerate. It is provable that no solution to the Byzantine Generals problem (which already assumes synchronicity, and known participants) can tolerate more than $(n - 1)/3$ byzantine faults, or 33% of the network acting maliciously [2]. This solution does not, however, require verifiable authenticity of the messages delivered between nodes (digital signatures). If a guarantee on the unforgeability of messages is possible, algorithms exist with much higher fault tolerance in the synchronous case.

Several algorithms with greater complexity have been proposed for Byzantine consensus in the asynchronous case. FaB Paxos [5] will tolerate $(n - 1)/5$ Byzantine failures in a network of n nodes, amounting to a tolerance of up to 20% of nodes in the network colluding maliciously. Attiya, Doyev, and Gill [3] introduce a phase algorithm for the asynchronous case, which can tolerate $(n - 1)/4$ failures, or up to 25% of the network. Lastly, Alchieri et al., 2008 [6] present BFT-CUP, which achieves Byzantine consensus in the asynchronous case even with unknown participants, with the maximal bound of a tolerance of $(n - 1)/3$ failures, but with additional restrictions on the connectivity of the underlying network.

2.4 Formal Consensus Goals

Our goal in this work is to show that the consensus algorithm utilized by the Ripple Protocol will achieve consensus at each ledger-close (even if consensus is the trivial consensus of all transactions being rejected), and that the trivial consensus will only be reached with a known probability, even in the face of Byzantine failures.

Since each node in the network only votes on proposals from a trusted set of nodes (the other nodes in its UNL), and since each node may have differing UNLs, we also show that only one consensus will be reached amongst all nodes, regardless of UNL membership. This goal is also referred to as preventing a “fork” in the network: a situation in which two disjoint sets of nodes each reach consensus independently, and two different last-closed ledgers are observed by nodes on each node-set.

Lastly we will show that the Ripple Protocol can achieve these goals in the face of $(n - 1)/5$ failures, which is not the strongest result in the literature, but we will also show that the Ripple Protocol possesses several other desirable features that greatly enhance its utility.

3. Ripple Consensus Algorithm

The Ripple Protocol consensus algorithm (RPCA), is applied every few seconds by all nodes, in order to maintain the correctness and agreement of the network. Once consensus is reached, the current ledger is considered “closed” and becomes the last-closed ledger. Assuming that the consensus algorithm is successful, and that there is no fork in the network, the last-closed ledger maintained by all nodes in the network will be identical.

3.1 Definition

The RPCA proceeds in rounds. In each round:

- Initially, each server takes all valid transactions it has seen prior to the beginning of the consensus round that have not already been applied (these may include new transactions initiated by end-users of the server, transactions held over from a previous consensus process, etc.), and makes them public in the form of a list known as the “candidate set”.
- Each server then amalgamates the candidate sets of all servers on its UNL, and votes on the veracity of all transactions.
- Transactions that receive more than a minimum percentage of “yes” votes are passed on to the next round, if there is one, while transactions that do not receive enough votes will either be discarded, or included in the candidate set for the beginning of the consensus process on the next ledger.
- The final round of consensus requires a minimum percentage of 80% of a server’s UNL agreeing

on a transaction. All transactions that meet this requirement are applied to the ledger, and that ledger is closed, becoming the new last-closed ledger.

3.2 Correctness

In order to achieve correctness, given a maximal amount of Byzantine failures, it must be shown that it is impossible for a fraudulent transaction to be confirmed during consensus, unless the number of faulty nodes exceeds that tolerance. The proof of the correctness of the RPCA then follows directly: since a transaction is only approved if 80% of the UNL of a server agrees with it, as long as 80% of the UNL is honest, no fraudulent transactions will be approved. Thus for a UNL of n nodes in the network, the consensus protocol will maintain correctness so long as:

$$f \leq (n - 1)/5 \quad (1)$$

where f is the number Byzantine failures. In fact, even in the face of $(n - 1)/5 + 1$ Byzantine failures, correctness is still technically maintained. The consensus process will fail, but it will still not be possible to confirm a fraudulent transaction. Indeed it would take $(4n + 1)/5$ Byzantine failures for an incorrect transaction to be confirmed. We call this second bound the bound for *weak* correctness, and the former the bound for *strong* correctness.

It should also be noted that not all “fraudulent” transactions pose a threat, even if confirmed during consensus. Should a user attempt to double-spend funds in two transactions, for example, even if both transactions are confirmed during the consensus process, after the first transaction is applied, the second will fail, as the funds are no longer available. This robustness is due to the fact that transactions are applied deterministically, and that consensus ensures that all nodes in the network are applying the deterministic rules to the same set of transactions.

For a slightly different analysis, let us assume that the probability that any node will decide to collude and join a nefarious cartel is p_c . Then the probability of correctness is given by p^* , where:

$$p^* = \sum_{i=0}^{\lceil (n-1)/5 \rceil} \binom{n}{i} p_c^i (1-p_c)^{n-i} \quad (2)$$

This probability represents the likelihood that the size of the nefarious cartel will remain below the maximal

threshold of Byzantine failures, given p_c . Since this likelihood is a binomial distribution, values of p_c greater than 20% will result in expected cartels of size greater than 20% of the network, thwarting the consensus process. In practice, a UNL is not chosen randomly, but rather with the intent to minimize p_c . Since nodes are not anonymous but rather cryptographically identifiable, selecting a UNL of nodes from a mixture of continents, nations, industries, ideologies, etc. will produce values of p_c much lower than 20%. As an example, the probability of the Anti-Defamation League and the Westboro Baptist Church colluding to defraud the network, is certainly much, much smaller than 20%. Even if the UNL has a relatively large p_c , say 15%, the probability of correctness is extremely high even with only 200 nodes in the UNL: 97.8%.

A graphical representation of how the probability of incorrectness scales as a function of UNL size for differing values of p_c is depicted in Figure 1. Note that here the vertical axis represents the probability of a nefarious cartel thwarting consensus, and thus lower values indicate greater probability of consensus success. As can be seen in the figure, even with a p_c as high as 10%, the probability of consensus being thwarted very quickly becomes negligible as the UNL grows past 100 nodes.

3.3 Agreement

To satisfy the agreement requirement, it must be shown that all nonfaulty nodes reach consensus on the same set of transactions, regardless of their UNLs. Since the UNLs for each server can be different, agreement is not inherently guaranteed by the correctness proof. For example, if there are no restrictions on the membership of the UNL, and the size of the UNL is not larger than $0.2 * n_{total}$ where n_{total} is the number of nodes in the entire network, then a fork is possible. This is illustrated by a simple example (depicted in figure 2): imagine two cliques within the UNL graph, each larger than $0.2 * n_{total}$. By cliques, we mean a set of nodes where each node's UNL is the selfsame set of nodes. Because these two cliques do not share any members, it is possible for each to achieve a correct consensus independently of each other, violating agreement. If the connectivity of the two cliques surpasses $0.2 * n_{total}$, then a fork is no longer possible, as disagreement between the cliques would prevent consensus from being reached at the 80% agreement threshold that is required.

An upper bound on the connectivity required to

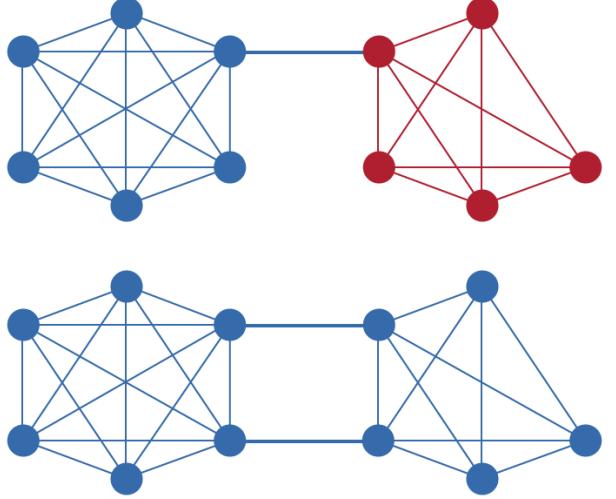


Figure 2. An example of the connectivity required to prevent a fork between two UNL cliques.

prove agreement is given by:

$$|UNL_i \cap UNL_j| \geq \frac{1}{5} \max(|UNL_i|, |UNL_j|) \forall i, j \quad (3)$$

This upper bound assumes a clique-like structure of UNLs, i.e. nodes form sets whose UNLs contain other nodes in those sets. This upper bound guarantees that no two cliques can reach consensus on conflicting transactions, since it becomes impossible to reach the 80% threshold required for consensus. A tighter bound is possible when indirect edges between UNLs are taken into account as well. For example, if the structure of the network is not clique-like, a fork becomes much more difficult to achieve, due to the greater entanglement of the UNLs of all nodes.

It is interesting to note that no assumptions are made about the nature of the intersecting nodes. The intersection of two UNLs may include faulty nodes, but so long as the size of the intersection is larger than the bound required to guarantee agreement, and the total number of faulty nodes is less than the bound required to satisfy strong correctness, then both correctness and agreement will be achieved. That is to say, agreement is dependent solely on the size of the intersection of nodes, not on the size of the intersection of nonfaulty nodes.

3.4 Utility

While many components of utility are subjective, one that is indeed provable is convergence: that the consensus process will terminate in finite time.

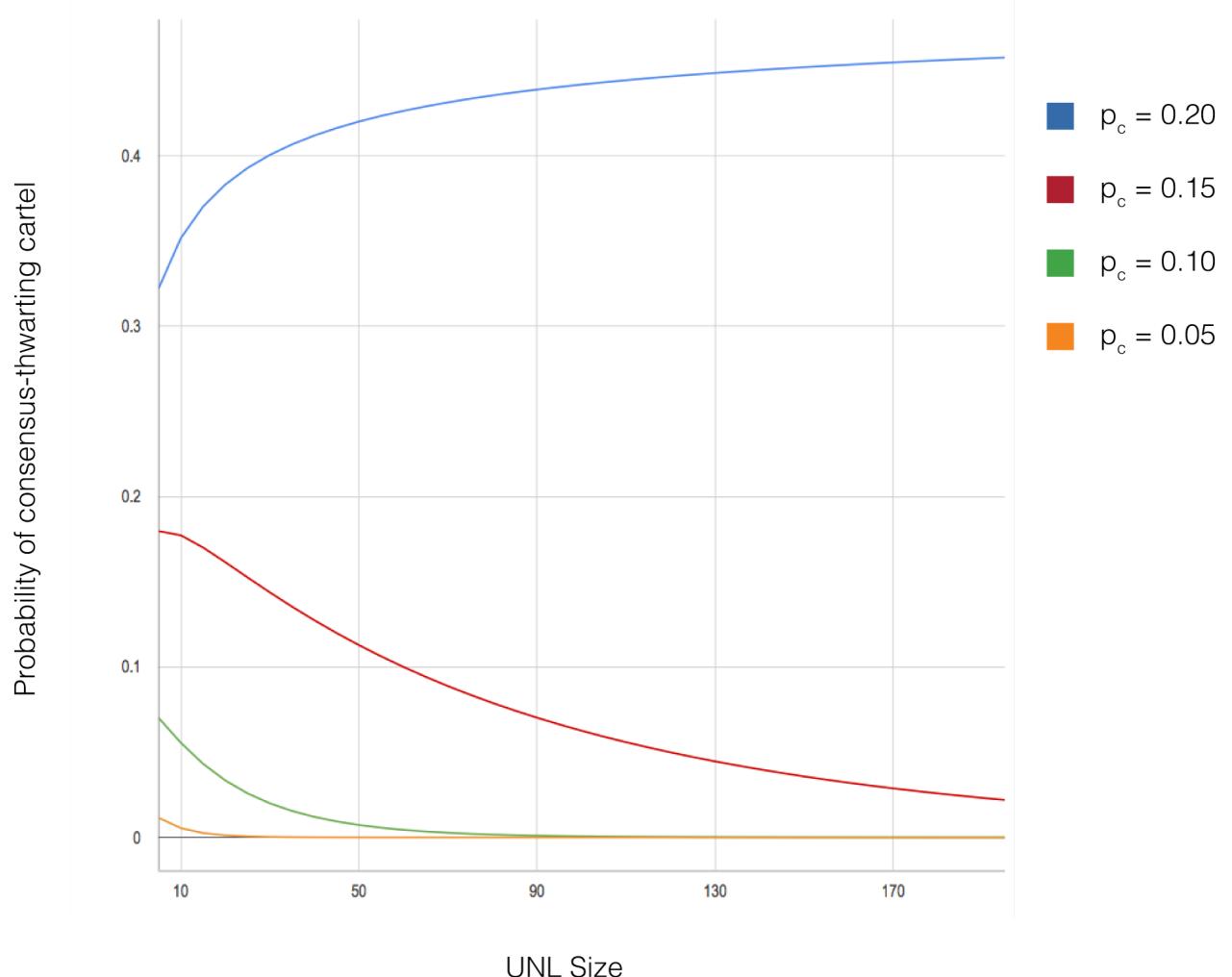


Figure 1. Probability of a nefarious cartel being able to thwart consensus as a function of the size of the UNL, for different values of p_c , the probability that any member of the UNL will decide to collude with others. Here, lower values indicate a higher probability of consensus success.

3.4.1 Convergence

We define convergence as the point in which the RPCA reaches consensus with strong correctness on the ledger, and that ledger then becomes the last-closed ledger. Note that while technically weak correctness still represents convergence of the algorithm, it is only convergence in the trivial case, as proposition **C3** is violated, and no transactions will ever be confirmed. From the results above, we know that strong correctness is always achievable in the face of up to $(n - 1)/5$ Byzantine failures, and that only one consensus will be achieved in the entire network so long as the UNL-connectedness condition is met (Equation 3). All that remains is to show that when both of these conditions are met, consensus is reached in finite time.

Since the consensus algorithm itself is deterministic, and has a preset number of rounds, t , before consensus is terminated, and the current set of transactions are declared approved or not-approved (even if at this point no transactions have more than the 80% required agreement, and the consensus is only the trivial consensus), the limiting factor for the termination of the algorithm is the communication latency between nodes. In order to bound this quantity, the response-time of nodes is monitored, and nodes whose latency grows larger than a preset bound b are removed from all UNLs. While this guarantees that consensus will terminate with an upper bound of tb , it is important to note that the bounds described for correctness and agreement above must be met by the *final* UNL, after all nodes that will be

dropped have been dropped. If the conditions hold for the initial UNLs for all nodes, but then some nodes are dropped from the network due to latency, the correctness and agreement guarantees do not automatically hold but must be satisfied by the new set of UNLs.

3.4.2 Heuristics and Procedures

As mentioned above, a latency bound heuristic is enforced on all nodes in the Ripple Network to guarantee that the consensus algorithm will converge. In addition, there are a few other heuristics and procedures that provide utility to the RPCA.

- There is a mandatory 2 second window for all nodes to propose their initial candidate sets in each round of consensus. While this does introduce a lower bound of 2 seconds to each consensus round, it also guarantees that all nodes with reasonable latency will have the ability to participate in the consensus process.
- As the votes are recorded in the ledger for each round of consensus, nodes can be flagged and removed from the network for some common, easily-identifiable malicious behaviors. These include nodes that vote “No” on every transaction, and nodes that consistently propose transactions which are not validated by consensus.
- A curated default UNL is provided to all users, which is chosen to minimize p_c , described in section 3.2. While users can and should select their own UNLs, this default list of nodes guarantees that even naive users will participate in a consensus process that achieves correctness and agreement with extremely high probability.
- A network split detection algorithm is also employed to avoid a fork in the network. While the consensus algorithm certifies that the transactions on the last-closed ledger are correct, it does not prohibit the possibility of more than one last-closed ledger existing on different subsections of the network with poor connectivity. To try and identify if such a split has occurred, each node monitors the size of the active members of its UNL. If this size suddenly drops below a preset threshold, it is possible that a split has occurred. In order to prevent a false positive in the case where a large section of a UNL has temporary latency, nodes are allowed to publish a “partial

validation”, in which they do not process or vote on transactions, but declare that are still participating in the consensus process, as opposed to a different consensus process on a disconnected subnetwork.

- While it would be possible to apply the RPCA in just one round of consensus, utility can be gained through multiple rounds, each with an increasing minimum-required percentage of agreement, before the final round with an 80% requirement. These rounds allow for detection of latent nodes in the case that a few such nodes are creating a bottleneck in the transaction rate of the network. These nodes will be able to initially keep up during the lower-requirement rounds but fall behind and be identified as the threshold increases. In the case of one round of consensus, it may be the case that so few transactions pass the 80% threshold, that even slow nodes can keep up, lowering the transaction rate of the entire network.

4. Simulation Code

The provided simulation code demonstrates a round of RPCA, with parameterizable features (the number of nodes in the network, the number of malicious nodes, latency of messages, etc.). The simulator begins in perfect disagreement (half of the nodes in the network initially propose “yes”, while the other half propose “no”), then proceeds with the consensus process, showing at each stage the number of yes/no votes in the network as nodes adjust their proposals based upon the proposals of their UNL members. Once the 80% threshold is reached, consensus is achieved. We encourage the reader to experiment with different values of the constants defined at the beginning of “Sim.cpp”, in order to become familiar with the consensus process under different conditions.

5. Discussion

We have described the RPCA, which satisfies the conditions of correctness, agreement, and utility which we have outlined above. The result is that the Ripple Protocol is able to process secure and reliable transactions in a matter of seconds: the length of time required for one round of consensus to complete. These transactions are provably secure up to the bounds outlined in section 3, which, while not the strongest available in the literature for Asynchronous Byzantine consensus, do

allow for rapid convergence and flexibility in network membership. When taken together, these qualities allow the Ripple Network to function as a fast and low-cost global payment network with well-understood security and reliability properties.

While we have shown that the Ripple Protocol is provably secure so long as the bounds described in equations 1 and 3 are met, it is worth noting that these are maximal bounds, and in practice the network may be secure under significantly less stringent conditions. It is also important to recognize, however, that satisfying these bounds is not inherent to the RPCA itself, but rather requires management of the UNLs of all users. The default UNL provided to all users is already sufficient, but should a user make changes to the UNL, it must be done with knowledge of the above bounds. In addition, some monitoring of the global network structure is required in order to ensure that the bound in equation 3 is met, and that agreement will always be satisfied.

We believe the RPCA represents a significant step forward for distributed payment systems, as the low-latency allows for many types of financial transactions previously made difficult or even impossible with other, higher latency consensus methods.

6. Acknowledgments

Ripple Labs would like to acknowledge all of the people involved in the development of the Ripple Protocol consensus algorithm. Specifically, Arthur Britto, for his work on transaction sets, Jed McCaleb, for the original Ripple Protocol consensus concept, and David Schwartz, for his work on the “failure to agree is agreement to defer” aspect of consensus. Ripple Labs would also like to acknowledge Noah Youngs for his efforts in preparing and reviewing this paper.

References

- [1] Nakamoto, Satoshi. “Bitcoin: A peer-to-peer electronic cash system.” Consulted 1.2012 (2008): 28.
- [2] Lamport, Leslie, Robert Shostak, and Marshall Pease. “The Byzantine generals problem.” ACM Transactions on Programming Languages and Systems (TOPLAS) 4.3 (1982): 382-401.
- [3] Attiya, C., D. Dolev, and J. Gill. “Asynchronous Byzantine Agreement.” Proc. 3rd. Annual ACM Symposium on Principles of Distributed Computing. 1984.

- [4] Fischer, Michael J., Nancy A. Lynch, and Michael S. Paterson. “Impossibility of distributed consensus with one faulty process.” Journal of the ACM (JACM) 32.2 (1985): 374-382.
- [5] Martin, J-P., and Lorenzo Alvisi. “Fast byzantine consensus.” Dependable and Secure Computing, IEEE Transactions on 3.3 (2006): 202-215.
- [6] Alchieri, Eduardo AP, et al. “Byzantine consensus with unknown participants.” Principles of Distributed Systems. Springer Berlin Heidelberg, 2008. 22-40.

The Stellar Consensus Protocol: A Federated Model for Internet-level Consensus

DAVID MAZIÈRES, Stellar Development Foundation

This paper introduces a new model for consensus called federated Byzantine agreement (FBA). FBA achieves robustness through quorum slices—individual trust decisions made by each node that together determine system-level quorums. Slices bind the system together much the way individual networks’ peering and transit decisions now unify the Internet.

We also present the Stellar Consensus Protocol (SCP), a construction for FBA. Like all Byzantine agreement protocols, SCP makes no assumptions about the rational behavior of attackers. Unlike prior Byzantine agreement models, which presuppose a unanimously accepted membership list, SCP enjoys open membership that promotes organic network growth. Compared to decentralized proof-of-work and proof-of-stake schemes, SCP has modest computing and financial requirements, lowering the barrier to entry and potentially opening up financial systems to new participants.

CCS Concepts: •**Security and privacy** → **Distributed systems security; Security protocols;**

Additional Key Words and Phrases: Byzantine fault tolerance, asynchronous systems

1. INTRODUCTION

Financial infrastructure is currently a mess of closed systems. Gaps between these systems mean that transaction costs are high [Provost 2013] and money moves slowly across political and geographic boundaries [Banning-Lover 2015; CGAP 2008]. This friction has curtailed the growth of financial services, leaving billions of people underserved financially [Demirguc-Kunt et al. 2015].

To solve these problems, we need financial infrastructure that supports the kind of organic growth and innovation we’ve seen from the Internet, yet still ensures the integrity of financial transactions. Historically, we have relied on high barriers to entry to ensure integrity. We trust established financial institutions and do our best to regulate them. But this exclusivity conflicts with the goal of organic growth. Growth demands new, innovative participants, who may possess only modest financial and computing resources.

We need a worldwide financial network open to anyone, so that new organizations can join and extend financial access to unserved communities. The challenge for such a network is ensuring participants record transactions correctly. With a low barrier to entry, users won’t trust providers to police themselves. With worldwide reach, providers won’t all trust a single entity to operate the network. A compelling alternative is a decentralized system in which participants together ensure integrity by agreeing on the validity of one another’s transactions. Such agreement hinges on a mechanism for worldwide consensus.

This paper presents federated Byzantine agreement (FBA), a model suitable for worldwide consensus. In FBA, each participant knows of others it considers important. It waits for the vast majority of those others to agree on any transaction before considering the transaction settled. In turn, those important participants do not agree to the transaction until the participants *they* consider important agree as well, and so on. Eventually, enough of the network accepts a transaction that it becomes infeasible for an attacker to roll it back. Only then do any participants consider the transaction settled. FBA’s consensus can ensure the integrity of a financial network. Its decentralized control can spur organic growth.

This paper further presents the Stellar consensus protocol (SCP), a construction for FBA. We prove that SCP’s safety is optimal for an asynchronous protocol, in that it guarantees agreement under any node-failure scenario that admits such a guarantee.

We also show that SCP is free from *blocked* states—in which consensus is no longer possible—unless participant failures make it impossible to satisfy trust dependencies. SCP is the first provably safe consensus mechanism to enjoy four key properties simultaneously:

- **Decentralized control.** Anyone is able to participate and no central authority dictates whose approval is required for consensus.
- **Low latency.** In practice, nodes can reach consensus at timescales humans expect for web or payment transactions—i.e., a few seconds at most.
- **Flexible trust.** Users have the freedom to trust any combination of parties they see fit. For example, a small non-profit may play a key role in keeping much larger institutions honest.
- **Asymptotic security.** Safety rests on digital signatures and hash families whose parameters can realistically be tuned to protect against adversaries with unimaginably vast computing power.

SCP has applications beyond financial markets for ensuring organizations perform important functions honestly. An example is certificate authorities (CAs), who literally hold the keys to the web. Experience shows that CAs sign incorrect certificates that get used in the wild [Microsoft 2013; Langley 2015]. Several proposals address this problem through certificate transparency [Kim et al. 2013; Laurie et al. 2013; Basin et al. 2014; Melara et al. 2014]. Certificate transparency allows users to examine the history of certificates issued for any given entity and detect attempts by CAs to change an entity’s public key without the endorsement of the previous key. SCP holds the potential to strengthen the indelible certificate history at the core of certificate transparency. Demanding global consensus on certificate history among a decentralized group of auditors would make it harder to backpedal and override previously issued certificates.

The next section discusses previous approaches to consensus. Section 3 defines federated Byzantine agreement (FBA) and lays out notions of safety and liveness applicable in the FBA model. Section 4 discusses optimal failure resilience in an FBA system, thereby establishing the security goals for SCP. Section 5 develops federated voting, a key building block of the SCP protocol. Section 6 presents SCP itself, proving safety and freedom from blocked states. Section 7 discusses limitations of SCP. Finally, Section 8 summarizes results. For readers less familiar with mathematical notation, Appendix A defines some symbols used throughout the paper.

2. RELATED WORK

Figure 1 summarizes how SCP differs from previous consensus mechanisms. The most famous decentralized consensus mechanism is the proof-of-work scheme advanced by Bitcoin [Nakamoto 2008]. Bitcoin takes a two-pronged approach to consensus. First, it provides incentives for rational actors to behave well. Second, it settles transactions through a proof-of-work [Dwork and Naor 1992] algorithm designed to protect against ill-behaved actors who do not possess the majority of the system’s computing power. Bitcoin has overwhelmingly demonstrated the appeal of decentralized consensus [Bonneau et al. 2015].

Proof of work has limitations, however. First, it wastes resources: by one estimate from 2014, Bitcoin might consume as much electric power as the entire country of Ireland [O’Dwyer and Malone 2014]. Second, secure transaction settlement suffers from expected latencies in the minutes or tens of minutes [Karame et al. 2012]. Finally, in contrast to traditional cryptographic protocols, proof of work offers no asymptotic security. Given non-rational attackers—or ones with extrinsic incentives to sabotage

mechanism	decentralized control	low latency	flexible trust	asymptotic security
proof of work	✓			
proof of stake	✓	maybe		maybe
Byzantine agreement		✓	✓	✓
Tendermint	✓	✓		✓
SCP (this work)	✓	✓	✓	✓

Fig. 1. Properties of different consensus mechanisms

consensus—small computational advantages can invalidate the security assumption, allowing history to be re-written in so-called “51% attacks.” Worse, attackers initially controlling less than 50% of computation can game the system to provide disproportionate rewards for those who join them [Eyal and Sirer 2013], thereby potentially gaining majority control. As the leading digital currency backed by the most computational power, Bitcoin enjoys a measure of protection against 51% attacks. Smaller systems have fallen victim [crazyearner 2013; Bradbury 2013], however, posing a problem for any proof-of-work system not built on the Bitcoin block chain.

An alternative to proof of work is proof of stake [King and Nadal 2012], in which consensus depends on parties that have posted collateral. Like proof of work, rewards encourage rational participants to obey the protocol; some designs additionally penalize bad behavior [Buterin 2014; Davarpanah et al. 2015]. Proof of stake opens the possibility of so-called “nothing at stake” attacks, in which parties that previously posted collateral but later cashed it in and spent the money can go back and rewrite history from a point where they still had stake. To mitigate such attacks, systems effectively combine proof of stake with proof of work—scaling down the required work in proportion to stake—or delay refunding collateral long enough for some other (sometimes informal) consensus mechanism to establish an irreversible checkpoint.

Still another approach to consensus is Byzantine agreement [Pease et al. 1980; Lamport et al. 1982], the best known variant of which is PBFT [Castro and Liskov 1999]. Byzantine agreement ensures consensus despite arbitrary (including non-rational) behavior on the part of some fraction of participants. This approach has two appealing properties. First, consensus can be fast and efficient. Second, trust is entirely decoupled from resource ownership, which makes it possible for a small non-profit to help keep more powerful organizations, such as banks or CAs, honest. Complicating matters, however, all parties must agree on the exact list of participants. Moreover, attackers must be prevented from joining multiple times and exceeding the system’s failure tolerance, a so-called Sybil attack [Douceur 2002]. BFT-CUP [Alchieri et al. 2008] accommodates unknown participants, but still presupposes a Sybil-proof centralized admission-control mechanism.

Generally, membership in Byzantine agreement systems is set by a central authority or closed negotiation. Prior attempts to decentralize admission have given up some of the benefits. One approach, taken by Ripple, is to publish a “starter” membership list that participants can edit for themselves, hoping people’s edits are either inconsequential or reproduced by an overwhelming fraction of participants. Unfortunately, because divergent lists invalidate safety guarantees [Schwartz et al. 2014], users are reluctant to edit the list in practice and a great deal of power ends up concentrated in the maintainer of the starter list. Another approach, taken by Tendermint [Kwon 2014], is to base membership on proof of stake. However, doing so once again ties trust to resource

ownership. SCP is the first Byzantine agreement protocol to give each participant maximum freedom in choosing which combinations of other participants to trust.

3. FEDERATED BYZANTINE AGREEMENT SYSTEMS

This section introduces the federated Byzantine agreement (FBA) model. Like non-federated Byzantine agreement, FBA addresses the problem of updating replicated state, such as a transaction ledger or certificate tree. By agreeing on what updates to apply, nodes avoid contradictory, irreconcilable states. We identify each update by a unique *slot* from which inter-update dependencies can be inferred. For instance, slots may be consecutively numbered positions in a sequentially applied log.

An FBA system runs a *consensus protocol* that ensures nodes agree on slot contents. A node v can safely apply update x in slot i when it has safely applied updates in all slots upon which i depends and, additionally, it believes all correctly functioning nodes will eventually agree on x for slot i . At this point, we say v has *externalized* x for slot i . The outside world may react to externalized values in irreversible ways, so a node cannot later change its mind about them.

A challenge for FBA is that malicious parties can join many times and outnumber honest nodes. Hence, traditional majority-based quorums do not work. Instead, FBA determines quorums in a decentralized way, by each node selecting what we call quorum slices. The next subsection defines quorums based on slices. The following subsection provides some examples and discussion. Finally, we define the key properties of safety and liveness that a consensus protocol should hope to achieve.

3.1. Quorum slices

In a consensus protocol, nodes exchange messages asserting statements about slots. We assume such assertions cannot be forged, which can be guaranteed if nodes are named by public key and they digitally sign messages. When a node hears a sufficient set of nodes assert a statement, it assumes no functioning node will ever contradict that statement. We call such a sufficient set a *quorum slice*, or, more concisely, just a *slice*. To permit progress in the face of node failures, a node may have multiple slices, any one of which is sufficient to convince it of a statement. At a high level, then, an FBA system consists of a loose confederation of nodes each of which has chosen one or more slices. More formally:

Definition (FBAS). A federated Byzantine agreement system, or *FBAS*, is a pair $\langle \mathbf{V}, \mathbf{Q} \rangle$ comprising a set of nodes \mathbf{V} and a quorum function $\mathbf{Q} : \mathbf{V} \rightarrow 2^{2^{\mathbf{V}}} \setminus \{\emptyset\}$ specifying one or more quorum slices for each node, where a node belongs to all of its own quorum slices—i.e., $\forall v \in \mathbf{V}, \forall q \in \mathbf{Q}(v), v \in q$. (Note 2^X denotes the powerset of X .)

Definition (quorum). A set of nodes $U \subseteq \mathbf{V}$ in FBAS $\langle \mathbf{V}, \mathbf{Q} \rangle$ is a *quorum* iff $U \neq \emptyset$ and U contains a slice for each member—i.e., $\forall v \in U, \exists q \in \mathbf{Q}(v)$ such that $q \subseteq U$.

A quorum is a set of nodes sufficient to reach agreement. A quorum slice is the subset of a quorum convincing one particular node of agreement. A quorum slice may be smaller than a quorum. Consider the four-node system in Figure 2, where each node has a single slice and arrows point to the other members of that slice. Node v_1 's slice $\{v_1, v_2, v_3\}$ is sufficient to convince v_1 of a statement. But v_2 's and v_3 's slices include v_4 , meaning neither v_2 nor v_3 can assert a statement without v_4 's agreement. Hence, no agreement is possible without v_4 's participation, and the only quorum including v_1 is the set of all nodes $\{v_1, v_2, v_3, v_4\}$.

Traditional, non-federated Byzantine agreement requires all nodes to accept the same slices, meaning $\forall v_1, v_2, \mathbf{Q}(v_1) = \mathbf{Q}(v_2)$. Because every member accepts every slice, traditional systems do not distinguish between slices and quorums. The downside is

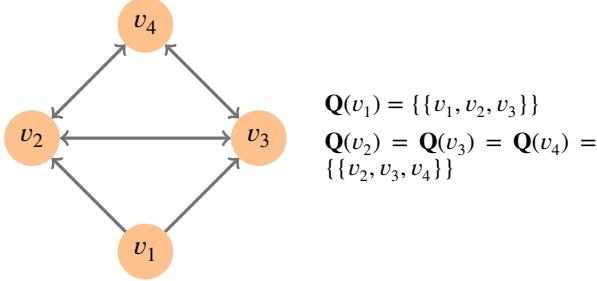
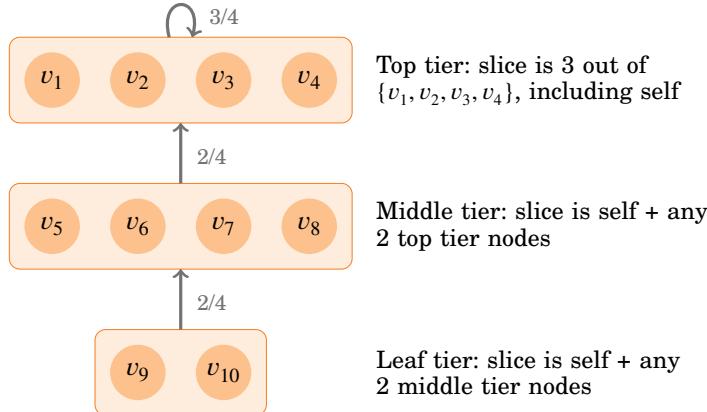
Fig. 2. v_1 's quorum slice is not a quorum without v_4 .

Fig. 3. Tiered quorum structure example

that membership and quorums must somehow be pre-ordained, precluding open membership and decentralized control. A traditional system, such as PBFT [Castro and Liskov 1999], typically has $3f + 1$ nodes, any $2f + 1$ of which constitute a quorum. Here f is the maximum number of Byzantine failures—meaning nodes acting arbitrarily—the system can survive.

FBA, introduced by this paper, generalizes Byzantine agreement to accommodate a greater range of settings. FBA's key innovation is enabling each node v to chose its own quorum slice set $Q(v)$. System-wide quorums thus arise from individual decisions made by each node. Nodes may select slices based on arbitrary criteria such as reputation or financial arrangements. In some settings, no individual node may have complete knowledge of all nodes in the system, yet consensus should still be possible.

3.2. Examples and discussion

Figure 3 shows an example of a tiered system in which different nodes have different slice sets, something possible only with FBA. A top tier, comprising v_1, \dots, v_4 , is structured like a PBFT system with $f = 1$, meaning it can tolerate one Byzantine failure so long as the other three nodes are reachable and well-behaved. Nodes v_5, \dots, v_8 constitute a middle tier and depend not on each other, but rather on the top tier. Only two top tier nodes are required to form a slice for a middle tier node. (The top tier assumes at most one Byzantine failure, so two top tier nodes cannot both fail unless the whole system has failed.) Nodes v_9 and v_{10} are in a leaf tier for which a slice consists of any

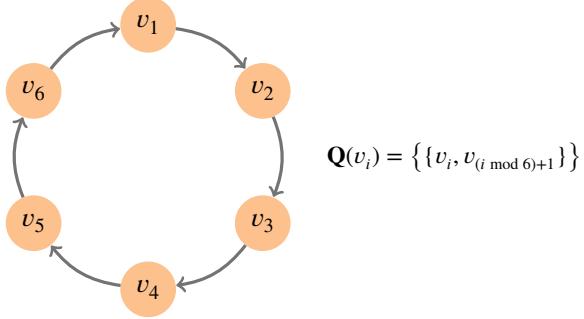


Fig. 4. Cyclic quorum structure example

two middle tier nodes. Note that v_9 and v_{10} may pick disjoint slices such as $\{v_5, v_6\}$ and $\{v_7, v_8\}$; nonetheless, both will indirectly depend on the top tier.

In practice, the top tier could consist of anywhere from four to dozens of widely known and trusted financial institutions. As the size of the top tier grows, there may not be exact agreement on its membership, but there will be significant overlap between most parties' notions of top tier. Additionally, one can imagine multiple middle tiers, for instance one for each country or geographic region.

This tiered structure resembles inter-domain network routing. The Internet today is held together by individual peering and transit relationships between pairs of networks. No central authority dictates or arbitrates these arrangements. Yet these pairwise relationships have sufficed to create a notion of *de facto* tier one ISPs [Norton 2010]. Though Internet reachability does suffer from firewalls, *transitive* reachability is nearly complete—e.g., a firewall might block The New York Times, but if it allows Google, and Google can reach The New York Times, then The New York Times is transitively reachable. Transitive reachability may be of limited utility for web sites, but it is crucial for consensus; the equivalent example would be Google accepting statements only if The New York Times does.

If we think of quorum slices as analogous to network reachability and quorums as analogous to transitive reachability, then the Internet's near complete transitive reachability suggests we can likewise ensure worldwide consensus with FBA. In many ways, consensus is an easier problem than inter-domain routing. While transit consumes resources and costs money, slice inclusion merely requires checking digital signatures. Hence, FBA nodes can err on the side of inclusiveness, constructing conservative slices with greater interdependence and redundancy than typically seen in peering and transit arrangements.

Another example not possible with centralized consensus is cyclic dependency structures, such as the one depicted in Figure 4. Such a cycle is unlikely to arise intentionally, but when individual nodes choose their own slices, it is possible for the overall system to end up embedding dependency cycles. The bigger point is that, compared to traditional Byzantine agreement, an FBA protocol must cope with a far wider variety of quorum structures.

3.3. Safety and liveness

We categorize nodes as either *well-behaved* or *ill-behaved*. A well-behaved node chooses sensible quorum slices (discussed further in Section 4.1) and obeys the protocol, including eventually responding to all requests. An ill-behaved node does not. Ill-behaved nodes suffer Byzantine failure, meaning they behave arbitrarily. For instance, an ill-

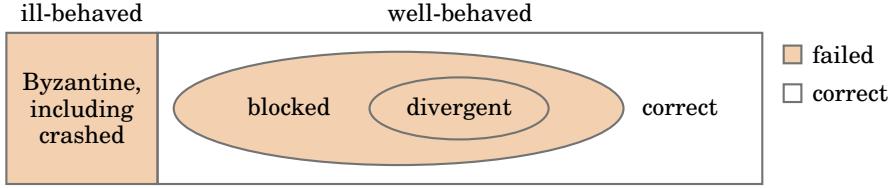


Fig. 5. Venn diagram of node failures

behaved node may be compromised, its owner may have maliciously modified the software, or it may have crashed.

The goal of Byzantine agreement is to ensure that well-behaved nodes externalize the same values despite the presence of such ill-behaved nodes. There are two parts to this goal. First, we would like to prevent nodes from diverging and externalizing different values for the same slot. Second, we would like to ensure nodes can actually externalize values, as opposed to getting blocked in some dead-end state from which consensus is no longer possible. We introduce the following two terms for these properties:

Definition (safety). A set of nodes in an FBAS enjoy *safety* if no two of them ever externalize different values for the same slot.

Definition (liveness). A node in an FBAS enjoys *liveness* if it can externalize new values without the participation of any failed (including ill-behaved) nodes.

We call well-behaved nodes that enjoy both safety and liveness *correct*. Nodes that are not correct have *failed*. All ill-behaved nodes have failed, but a well-behaved node can fail, too, by waiting indefinitely for messages from ill-behaved nodes, or, worse, by having its state poisoned by incorrect messages from ill-behaved nodes.

Figure 5 illustrates the possible kinds of node failure. To the left are Byzantine failures, meaning the ill-behaved nodes. To the right are two kinds of well-behaved but failed nodes. Nodes that lack liveness are termed *blocked*, while those that lack safety are termed *divergent*. An attack violating safety is strictly more powerful than one violating only liveness, so we classify divergent nodes as a subset of blocked ones.

Our definition of liveness is weak in that it says a node *can* externalize new values, not that it *will*. Hence, it admits a state of *perpetual preemption* in which consensus remains forever possible, yet the network continually thwarts it by delaying or re-ordering critical messages in just the wrong way. Perpetual preemption is inevitable in a purely asynchronous, deterministic system that survives node failure [Fischer et al. 1985]. Fortunately, preemption is transient. It does not indicate node failure, because the system can recover at any time. Protocols can mitigate the problem through randomness [Ben-Or 1983; Bracha and Toueg 1985] or through realistic assumptions about message latency [Dwork et al. 1988]. Latency assumptions are more practical when one would like to limit execution time or avoid the trusted dealers often required by more efficient Randomized algorithms [?]. Of course, only termination and not safety should depend upon message timing.

4. OPTIMAL RESILIENCE

Whether or not nodes enjoy safety and liveness depends on several factors: what quorum slices they have chosen, which nodes are ill-behaved, and of course the concrete consensus protocol and network behavior. As is common for asynchronous systems, we assume the network eventually delivers messages between well-behaved nodes, but can otherwise arbitrarily delay or reorder messages.

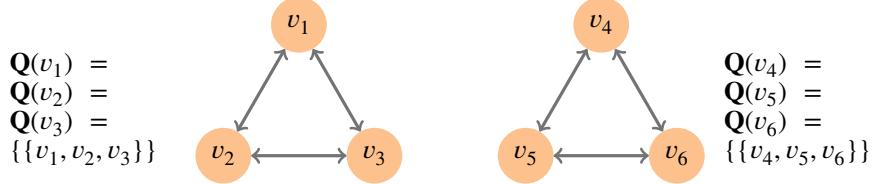
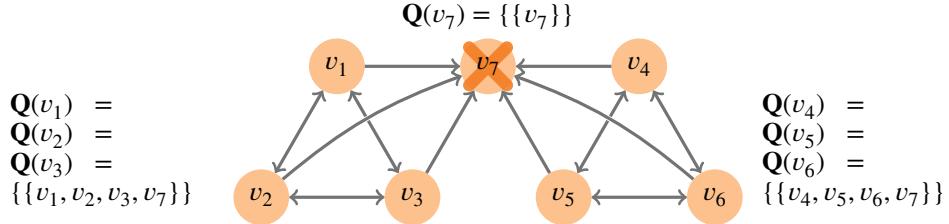


Fig. 6. FBAS lacking quorum intersection

Fig. 7. Ill-behaved node v_7 can undermine quorum intersection.

This section answers the following question: given a specific $\langle V, Q \rangle$ and particular subset of V that is ill-behaved, what are the best safety and liveness that any federated Byzantine agreement protocol can guarantee regardless of the network? We first discuss quorum intersection, a property without which safety is impossible to guarantee. We then introduce a notion of dispensable sets—sets of failed nodes in spite of which it is possible to guarantee both safety and liveness.

4.1. Quorum intersection

A protocol can guarantee agreement only if the quorum slices represented by function Q satisfy a validity property we call quorum intersection.

Definition (quorum intersection). An FBAS enjoys *quorum intersection* iff any two of its quorums share a node—i.e., for all quorums U_1 and U_2 , $U_1 \cap U_2 \neq \emptyset$.

Figure 6 illustrates a system lacking quorum intersection, where Q permits two quorums, $\{v_1, v_2, v_3\}$ and $\{v_4, v_5, v_6\}$, that do not intersect. Disjoint quorums can independently agree on contradictory statements, undermining system-wide agreement. When many quorums exist, quorum intersection fails if any two do not intersect. For example, the set of all nodes $\{v_1, \dots, v_6\}$ in Figure 6 is a quorum that intersects the other two, but the system still lacks quorum intersection because the other two do not intersect each other.

No protocol can guarantee safety in the absence of quorum intersection, since such a configuration can operate as two different FBAS systems that do not exchange any messages. However, even with quorum intersection, safety may be impossible to guarantee in the presence of ill-behaved nodes. Compare Figure 6, in which there are two disjoint quorums, to Figure 7, in which two quorums intersect at a single node v_7 , and v_7 is ill-behaved. If v_7 makes inconsistent statements to the left and right quorums, the effect is equivalent to disjoint quorums.

In fact, since ill-behaved nodes contribute nothing to safety, no protocol can guarantee safety without the well-behaved nodes enjoying quorum intersection on their own. After all, in a worst-case scenario for safety, ill-behaved nodes can just always make any possible (contradictory) statement that completes a quorum. Two quorums overlapping only at ill-behaved nodes will again be able to operate like two different FBAS

systems thanks to the duplicity of the ill-behaved nodes. In short, FBAS $\langle V, Q \rangle$ can survive Byzantine failure by a set of nodes $B \subseteq V$ iff $\langle V, Q \rangle$ enjoys quorum intersection after deleting the nodes in B from V and from all slices in Q . More formally:

Definition (delete). If $\langle V, Q \rangle$ is an FBAS and $B \subseteq V$ is a set of nodes, then to *delete* B from $\langle V, Q \rangle$, written $\langle V, Q \rangle^B$, means to compute the modified FBAS $\langle V \setminus B, Q^B \rangle$ where $Q^B(v) = \{ q \setminus B \mid q \in Q(v) \}$.

It is the responsibility of each node v to ensure $Q(v)$ does not violate quorum intersection. One way to do so is to pick conservative slices that lead to large quorums. Of course, a malicious v may intentionally pick $Q(v)$ to violate quorum intersection. But a malicious v can also lie about the value of $Q(v)$ or ignore $Q(v)$ to make arbitrary assertions. In short, $Q(v)$'s value is not meaningful when v is ill-behaved. This is why the necessary property for safety—quorum intersection of well-behaved nodes after deleting ill-behaved nodes—is unaffected by the slices of ill-behaved nodes.

Suppose Figure 6 evolved from a three-node FBAS v_1, v_2, v_3 with quorum intersection to a six-node FBAS without. When v_4, v_5, v_6 join, they maliciously choose slices that violate quorum intersection and no protocol can guarantee safety for V . Fortunately, deleting the bad nodes to yield $\langle V, Q \rangle^{\{v_4, v_5, v_6\}}$ restores quorum intersection, meaning at least $\{v_1, v_2, v_3\}$ can enjoy safety. Note that deletion is conceptual, for the sake of describing optimal safety. A protocol should guarantee safety for v_1, v_2, v_3 without their needing to know that v_4, v_5, v_6 are ill-behaved.

4.2. Dispensable sets (DSets)

We capture the fault tolerance of nodes' slice selections through the notion of a *dispensable set* or *DSet*. Informally, the safety and liveness of nodes outside a DSet can be guaranteed regardless of the behavior of nodes inside the DSet. Put another way, in an optimally resilient FBAS, if a single DSet encompasses every ill-behaved node, it also contains every failed node, and conversely all nodes outside the DSet are correct. As an example, in a centralized PBFT system with $3f + 1$ nodes and quorum size $2f + 1$, any f or fewer nodes constitute a DSet. Since PBFT in fact survives up to f Byzantine failures, its robustness is optimal.

In the less regular example of Figure 3, $\{v_1\}$ is a DSet, since one top tier node can fail without affecting the rest of the system. $\{v_9\}$ is also a DSet because no other node depends on v_9 for correctness. $\{v_6, \dots, v_{10}\}$ is a DSet, because neither v_5 nor the top tier depend on any of those five nodes. $\{v_5, v_6\}$ is *not* a DSet, as it is a slice for v_9 and v_{10} and hence, if entirely malicious, can lie to v_9 and v_{10} and convince them of assertions inconsistent with each other or the rest of the system.

To prevent a misbehaving DSet from affecting the correctness of other nodes, two properties must hold. For safety, deleting the DSet cannot undermine quorum intersection. For liveness, the DSet cannot deny other nodes a functioning quorum. This leads to the following definition:

Definition (DSet). Let $\langle V, Q \rangle$ be an FBAS and $B \subseteq V$ be a set of nodes. We say B is a dispensable set, or *DSet*, iff:

- (1) (*quorum intersection despite B*) $\langle V, Q \rangle^B$ enjoys quorum intersection, and
- (2) (*quorum availability despite B*) Either $V \setminus B$ is a quorum in $\langle V, Q \rangle$ or $B = V$.

Quorum availability despite B protects against nodes in B refusing to answer requests and blocking other nodes' progress. Quorum intersection despite B protects against the opposite—nodes in B making contradictory assertions that enable other nodes to externalize inconsistent values for the same slot. Nodes must balance the two threats in slice selection. All else equal, bigger slices lead to bigger quorums with

well-behaved / ill-behaved	Local property of nodes, independent of other nodes (except for the validity of slice selection).
intact / befouled	Property of nodes given their quorum slices and a particular set of ill-behaved nodes. Befouled nodes are ill-behaved or depend, possibly indirectly, on too many ill-behaved nodes.
correct / failed	Property of nodes given their quorum slices, a concrete protocol, and actual network behavior. The goal of a consensus protocol is to guarantee correctness for all intact nodes.

Fig. 8. Key properties of FBAS nodes

greater overlap, meaning fewer failed node sets B will undermine quorum intersection when deleted. On the other hand, bigger slices are more likely to contain failed nodes, endangering quorum availability.

The smallest DSet containing all ill-behaved nodes may encompass well-behaved nodes as well, reflecting the fact that a sufficiently large set of ill-behaved nodes can cause well-behaved nodes to fail. For instance, in Figure 3, the smallest DSet containing v_5 and v_6 is $\{v_5, v_6, v_9, v_{10}\}$. The set of all nodes, V , is always a DSet, as an FBAS $\langle V, Q \rangle$ vacuously enjoys quorum intersection despite V and, by special case, also enjoys quorum availability despite V . The motivation for the special case is that given sufficiently many ill-behaved nodes, V may be the smallest DSet to contain all ill-behaved ones, indicating a scenario under which no protocol can guarantee anything better than complete system failure.

The DSets in an FBAS are determined *a priori* by the quorum function Q . Which nodes are well- and ill-behaved depends on runtime behavior, such as machines getting compromised. The DSets we care about are those that encompass all ill-behaved nodes, as they help us distinguish nodes that should be guaranteed correct from ones for which such a guarantee is impossible. To this end, we introduce the following terms:

Definition (intact). A node v in an FBAS is *intact* iff there exists a DSet B containing all ill-behaved nodes and such that $v \notin B$.

Definition (befouled). A node v in an FBAS is *befouled* iff it is not intact.

A befouled node v is surrounded by enough failed nodes to block its progress or poison its state, even if v itself is well-behaved. No FBAS can guarantee the correctness of a befouled node. However, an optimal FBAS guarantees that every intact node remains correct. Figure 8 summarizes the key properties of nodes. The following theorems facilitate analysis by showing that the set of befouled nodes is always a DSet in an FBAS with quorum intersection.

THEOREM 1. *Let U be a quorum in FBAS $\langle V, Q \rangle$, let $B \subseteq V$ be a set of nodes, and let $U' = U \setminus B$. If $U' \neq \emptyset$ then U' is a quorum in $\langle V, Q \rangle^B$.*

PROOF. Because U is a quorum, every node $v \in U$ has a $q \in Q(v)$ such that $q \subseteq U$. Since $U' \subseteq U$, it follows that every $v \in U'$ has a $q \in Q(v)$ such that $q \setminus B \subseteq U'$. Rewriting with deletion notation yields $\forall v \in U', \exists q \in Q^B(v)$ such that $q \subseteq U'$, which, because $U' \subseteq V \setminus B$, means that U' is a quorum in $\langle V, Q \rangle^B$. \square

THEOREM 2. *If B_1 and B_2 are DSets in an FBAS $\langle V, Q \rangle$ enjoying quorum intersection, then $B = B_1 \cap B_2$ is a DSet, too.*

PROOF. Let $U_1 = V \setminus B_1$ and $U_2 = V \setminus B_2$. If $U_1 = \emptyset$, then $B_1 = V$ and $B = B_2$ (a DSet), so we are done. Similarly, if $U_2 = \emptyset$, then $B = B_1$, and we are done. Otherwise, note

that by quorum availability despite DSets B_1 and B_2 , U_1 and U_2 are quorums in $\langle V, Q \rangle$. It follows from the definition that the union of two quorums is also a quorum. Hence $V \setminus B = U_1 \cup U_2$ is a quorum and we have quorum availability despite B .

We must now show quorum intersection despite B . Let U_a and U_b be any two quorums in $\langle V, Q \rangle^B$. Let $U = U_1 \cap U_2 = U_2 \setminus B_1$. By quorum intersection of $\langle V, Q \rangle$, $U = U_1 \cap U_2 \neq \emptyset$. But then by Theorem 1, $U = U_2 \setminus B_1$ must be a quorum in $\langle V, Q \rangle^{B_1}$. Now consider that $U_a \setminus B_1$ and $U_a \setminus B_2$ cannot both be empty, or else $U_a \setminus B = U_a$ would be. Hence, by Theorem 1, either $U_a \setminus B_1$ is a quorum in $(\langle V, Q \rangle^B)^{B_1} = \langle V, Q \rangle^{B_1}$, or $U_a \setminus B_2$ is a quorum in $(\langle V, Q \rangle^B)^{B_2} = \langle V, Q \rangle^{B_2}$, or both. In the former case, note that if $U_a \setminus B_1$ is a quorum in $\langle V, Q \rangle^{B_1}$, then by quorum intersection of $\langle V, Q \rangle^{B_1}$, $(U_a \setminus B_1) \cap U \neq \emptyset$; since $(U_a \setminus B_1) \cap U = (U_a \setminus B_1) \setminus B_2$, it follows that $U_a \setminus B_2 \neq \emptyset$, making $U_a \setminus B_2$ a quorum in $\langle V, Q \rangle^{B_2}$. By a similar argument, $U_b \setminus B_2$ must be a quorum in $\langle V, Q \rangle^{B_2}$. But then quorum intersection despite B_2 tells us that $(U_a \setminus B_2) \cap (U_b \setminus B_2) \neq \emptyset$, which is only possible if $U_a \cap U_b \neq \emptyset$. \square

THEOREM 3. *In an FBAS with quorum intersection, the set of befouled nodes is a DSet.*

PROOF. Let B_{\min} be the intersection of every DSet that contains all ill-behaved nodes. It follows from the definition of *intact* that a node v is intact iff $v \notin B_{\min}$. Thus, B_{\min} is precisely the set of befouled nodes. By Theorem 2, DSets are closed under intersection, so B_{\min} is also a DSet. \square

5. FEDERATED VOTING

This section develops a federated voting technique that FBAS nodes can use to agree on a statement. At a high level, the process for agreeing on some statement a involves nodes exchanging two sets of messages. First, nodes *vote* for a . Then, if the vote was successful, nodes *confirm* a , effectively holding a second vote on the fact that the first vote succeeded.

From each node's perspective, the two rounds of messages divide agreement on a statement a into three phases: unknown, accepted, and confirmed. (This pattern dates back to three-phase commit [Skeen and Stonebraker 1983].) Initially, a 's status is completely *unknown* to a node v — a could end up true, false, or even *stuck* in a permanently indeterminate state. If the first vote succeeds, v may come to *accept* a . No two intact nodes ever accept contradictory statements, so if v is intact and accepts a , then a cannot be false.

For two reasons, however, v accepting a does not suffice for v to act on a . First, the fact that v accepted a does not mean all intact nodes can; a could be stuck for other nodes. Second, if v is befouled, then accepting a means nothing— a may be false at intact nodes. Yet even if v is befouled—which v does not know—the system may still enjoy quorum intersection of well-behaved nodes, in which case, for optimal safety, v needs greater assurance of a . Holding a second vote addresses both problems. If the second vote succeeds, v moves to the *confirmed* phase in which it can finally deem a true and act on it.

The next few subsections detail the federated voting process. Because voting does not rule out the possibility of stuck statements, Section 5.6 discusses how to cope with them. Section 6 will turn federated voting into a consensus protocol that avoids the possibility of stuck slots for intact nodes.

5.1. Voting with open membership

A correct node in a Byzantine agreement system acts on a statement a only when it knows that other correct nodes will never agree to statements contradicting a . Most protocols employ voting for this purpose. Well-behaved nodes vote for a statement a only if it is valid. Well-behaved nodes also never change their votes. Hence, in centralized Byzantine agreement, it is safe to accept a if a quorum comprising a majority of well-behaved nodes has voted for it. We say a statement is *ratified* once it has received the necessary votes.

In a federated setting, we must adapt voting to accommodate open membership. One difference is that a quorum no longer corresponds to a majority of well-behaved nodes. However, the majority requirement primarily serves to ensure quorum intersection of well-behaved nodes, which Section 4.1 already adapted to FBA. Another implication of open membership is that nodes must discover what constitutes a quorum as part of the voting process. To implement quorum discovery, a protocol should specify $Q(v)$ in all messages from v .

Definition (vote). A node v votes for an (abstract) statement a iff

- (1) v asserts a is valid and consistent with all statements v has accepted, and
- (2) v asserts it has never voted against a —i.e., voted for a statement that contradicts a —and v promises never to vote against a in the future.

Definition (ratify). A quorum U_a ratifies a statement a iff every member of U_a votes for a . A node v ratifies a iff v is a member of a quorum U_a that ratifies a .

THEOREM 4. *Two contradictory statements a and \bar{a} cannot both be ratified in an FBAS that enjoys quorum intersection and contains no ill-behaved nodes.*

PROOF. By contradiction. Suppose quorum U_1 ratifies a and quorum U_2 ratifies \bar{a} . By quorum intersection, $\exists v \in U_1 \cap U_2$. Such a v must have illegally voted for both a and \bar{a} , violating the assumption of no ill-behaved nodes. \square

THEOREM 5. *Let $\langle V, Q \rangle$ be an FBAS enjoying quorum intersection despite B , and suppose B contains all ill-behaved nodes. Let v_1 and v_2 be two nodes not in B . Let a and \bar{a} be contradictory statements. If v_1 ratifies a then v_2 cannot ratify \bar{a} .*

PROOF. By contradiction. Suppose v_1 ratifies a and v_2 ratifies \bar{a} . By definition, there must exist a quorum U_1 containing v_1 that ratified a and quorum U_2 containing v_2 that ratified \bar{a} . By Theorem 1, since $U_1 \setminus B \neq \emptyset$ and $U_2 \setminus B \neq \emptyset$, both must be quorums in $\langle V, Q \rangle^B$, meaning they ratified a and \bar{a} respectively in $\langle V, Q \rangle^B$. But $\langle V, Q \rangle^B$ enjoys quorum intersection and has no ill-behaved nodes, so Theorem 4 tell us a and \bar{a} cannot both be ratified. \square

THEOREM 6. *Two intact nodes in an FBAS with quorum intersection cannot ratify contradictory statements.*

PROOF. Let B be the set of befouled nodes. By Theorem 3, B is a DSet. By the definition of DSet, $\langle V, Q \rangle$ enjoys quorum intersection despite B . By Theorem 5, two nodes not in B cannot ratify contradictory statements. \square

5.2. Blocking sets

In centralized consensus, liveness is an all-or-nothing property of the system. Either a unanimously well-behaved quorum exists, or else ill-behaved nodes can prevent the rest of the system from accepting new statements. In FBA, by contrast, liveness may differ across nodes. For instance, in the tiered quorum example of Figure 3, if middle

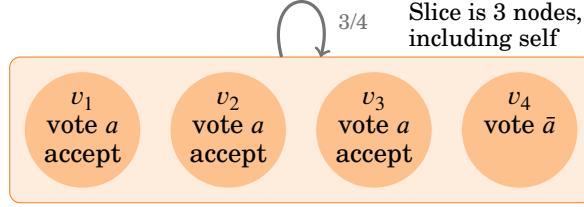


Fig. 9. v_4 voted for \bar{a} , which contradicts ratified statement a .

tier nodes v_6, v_7, v_8 crash, the leaf tier will be blocked while the top tier and node v_5 will continue to enjoy liveness.

An FBA protocol can guarantee liveness to a node v only if $Q(v)$ contains at least one quorum slice comprising only correct nodes. A set B of failed nodes can violate this property if B contains at least one member of each of v 's slices. We term such a set B v -blocking, because it has the power to block progress by v .

Definition (v -blocking). Let $v \in V$ be a node in FBAS $\langle V, Q \rangle$. A set $B \subseteq V$ is v -blocking iff it overlaps every one of v 's slices—i.e., $\forall q \in Q(v), q \cap B \neq \emptyset$.

THEOREM 7. Let $B \subseteq V$ be a set of nodes in FBAS $\langle V, Q \rangle$. $\langle V, Q \rangle$ enjoys quorum availability despite B iff B is not v -blocking for any $v \in V \setminus B$.

PROOF. “ $\forall v \in V \setminus B, B$ is not v -blocking” is equivalent to “ $\forall v \in V \setminus B, \exists q \in Q(v)$ such that $q \subseteq V \setminus B$.” By the definition of *quorum*, the latter holds iff $V \setminus B$ is a quorum or $B = V$, the exact definition of *quorum availability despite B* . \square

As a corollary, the DSet of befouled nodes is not v -blocking for any intact v .

5.3. Accepting statements

When an intact node v learns that it has ratified a statement, Theorem 6 tells v that other intact nodes will not ratify contradictory statements. This condition is sufficient for v to accept a , but we cannot make it necessary. Ratifying a statement requires voting for it, and some nodes may have voted for contradictory statements. In Figure 9, for example, v_4 votes for \bar{a} before learning that the other three nodes ratified the contradictory statement a . Though v_4 cannot now vote for a , we would still like it to accept a to be consistent with the other nodes.

A key insight is that if a node v is intact, then no v -blocking set B can consist entirely of befouled nodes. Now suppose B is a v -blocking set and every member of B claims to accept statement a . If v is intact, at least one member of B must be, too. The intact member will not lie about accepting a ; hence, a is true and v can accept it. Of course, if v is befouled, then a might not be true. But a befouled node can accept anything and vacuously not affect the correctness of intact nodes.

Definition (accept). An FBAS node v accepts a statement a iff it has never accepted a statement contradicting a and it determines that either

- (1) There exists a quorum U such that $v \in U$ and each member of U either voted for a or claims to accept a , or
- (2) Each member of a v -blocking set claims to accept a .

Though a well-behaved node cannot vote for contradictory statements, condition 2 above allows a node to *vote* for one statement and later *accept* a contradictory one.

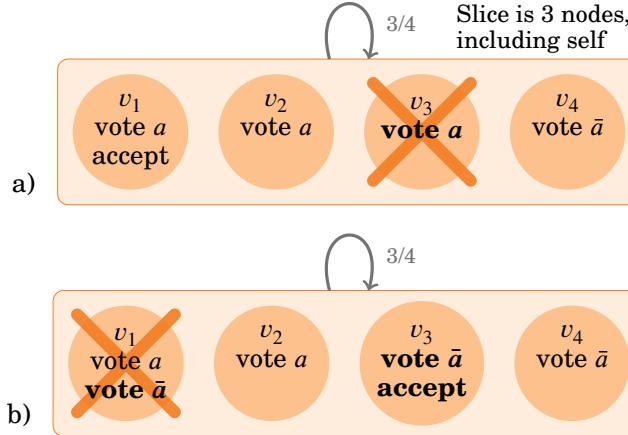


Fig. 10. Scenarios indistinguishable to v_2 when v_2 does not see bold messages

THEOREM 8. *Two intact nodes in an FBAS that enjoys quorum intersection cannot accept contradictory statements.*

PROOF. Let $\langle V, Q \rangle$ be an FBAS with quorum intersection and let B be its DSet of befouled nodes (which exists by Theorem 3). Suppose an intact node accepts statement a . Let v be the first intact node to accept a . At the point v accepts a , only befouled nodes in B can claim to accept it. Since by the corollary to Theorem 7, B cannot be v -blocking, it must be that v accepted a through condition 1. Thus, v identified a quorum U such that every node claimed to vote for or accept a , and since v is the first intact node to accept a , it must mean all nodes in $U \setminus B$ voted for a . In other words, v ratified a in $\langle V, Q \rangle^B$. Generalizing, any statement accepted by an intact node in $\langle V, Q \rangle$ must be ratified in $\langle V, Q \rangle^B$. Because B is a DSet, $\langle V, Q \rangle^B$ enjoys quorum intersection. Because additionally B contains all ill-behaved nodes, Theorem 4 rules out ratification of contradictory statements. \square

5.4. Accepting is not enough

Unfortunately, for nodes to assume the truth of accepted statements would yield sub-optimal safety and liveness guarantees in a federated consensus protocol. We discuss the issues with safety and liveness in turn. To provide some context, we then explain why these issues are thornier in FBA than in centralized Byzantine agreement.

5.4.1. Safety. Consider an FBAS $\langle V, Q \rangle$ in which the only quorum is unanimous consent—i.e., $\forall v, Q(v) = \{V\}$. This ought to be a conservative choice for safety—don’t do anything unless everyone agrees. Yet since every node is v -blocking for every v , any node can single-handedly convince any other node to accept arbitrary statements.

The problem is that accepted statements are only safe among intact nodes. But as discussed in Section 4.1, the only condition necessary to guarantee safety is quorum intersection of well-behaved nodes, which might hold even in the case that some well-behaved nodes are befouled. In particular, when $Q(v) = \{V\}$, the only DSets are \emptyset and V , meaning any node failure befouls the whole system. By contrast, quorum intersection holds despite every $B \subseteq V$.

5.4.2. Liveness. Another limitation of accepted statements is that other intact nodes may be unable to accept them. This possibility makes reliance on accepted statements

problematic for liveness. If a node proceeds to act on a statement because it accepted the statement, other nodes could be unable to proceed in a similar fashion.

Consider Figure 10a, in which node v_3 crashes after helping v_1 ratify and accept statement a . Though v_1 accepts a , v_2 and v_4 cannot. In particular, from v_2 's perspective, the situation depicted is indistinguishable from Figure 10b, in which v_3 voted for \bar{a} and is well-behaved but slow to respond, while v_1 is ill-behaved and sent v_3 a vote for \bar{a} (thereby causing v_3 to accept \bar{a}) while illegally also sending v_2 a vote for a .

To support a protocol-level notion of liveness in cases like Figure 10a, v_1 needs a way to ensure every other intact node can eventually accept a before v_1 acts on a . Once this is the case, it makes sense to say the system agrees on a .

Definition (agree). An FBAS $\langle V, Q \rangle$ *agrees* on a statement a iff, regardless of what subsequently transpires, once sufficient messages are delivered and processed, every intact node will accept a .

5.4.3. Comparison to centralized voting. To understand why the above issues arise in federated voting, consider a centralized Byzantine agreement system of N nodes with quorum size T . Such a system enjoys quorum availability with $f_L = N - T$ or fewer node failures. Since any two quorums share at least $2T - N$ nodes, quorum intersection of well-behaved nodes holds up to $f_S = 2T - N - 1$ Byzantine failures.

Centralized Byzantine agreement systems typically set $N = 3f + 1$ and $T = 2f + 1$ to yield $f_L = f_S = f$, the equilibrium point at which safety and liveness have the same fault tolerance. If safety is more important than liveness, some protocols increase T so that $f_S > f_L$ [Li and Mazières 2007]. In FBA, because quorums arise organically, systems are unlikely to find themselves at equilibrium, making it far more important to protect safety in the absence of liveness.

Now consider a centralized system in which, because of node failure and contradictory votes, some node v cannot ratify statement a that was ratified by other nodes. If v hears $f_S + 1$ nodes claim a was ratified, v knows that either one of them is well-behaved or all safety guarantees have collapsed. Either way, v can act on a with no loss of safety. The FBA equivalent would be to hear from a set B where B , if deleted, undermines quorum intersection of well-behaved nodes. Identifying such a B is hard for three reasons: one, quorums are discovered dynamically; two, ill-behaved nodes may lie about slices; and three, v does not know which nodes are well-behaved. Instead, we defined federated voting to accept a when a v -blocking set does. The v -blocking property has the advantage of being easily checkable, but is equivalent to hearing from $f_L + 1$ nodes in a centralized system when we really want $f_S + 1$.

To guarantee agreement among all well-behaved nodes in a centralized system, one merely needs $f_L + f_S + 1$ nodes to acknowledge that a statement was ratified. If more than f_L of them fail, we do not expect liveness anyway. If f_L or fewer fail, then we know $f_S + 1$ nodes remain willing to attest to ratification, which will in turn convince all other well-behaved nodes. The reliance on f_S has no easy analogue in the FBA model. Interestingly, however, $f_L + f_S + 1 = T$, the quorum size, suggesting a similar approach might work with a more complex justification.

Put another way, at some point nodes need to believe a statement strongly enough to depend on its truth for safety. A centralized system offers two ways to reach this point for a statement a : ratify a first-hand, or reason backwards from $f_S + 1$ nodes claiming a was ratified, figuring safety is hopeless if they have all lied. FBA lacks the latter approach; the only tool it has for safety among well-behaved nodes is first-hand ratification. Since nodes still need a way to overcome votes against ratified statements, we introduced a notion of accepting, but it provides a weaker consistency guarantee limited to intact nodes.

5.5. Statement confirmation

Both limitations of accepted statements stem from complications when a set of intact nodes S votes against a statement a that is nonetheless ratified. Particularly in light of FBA's non-uniform quorums, S may prevent some intact node from ever ratifying v . To provide v a means of accepting a despite votes against it, the definition of *accept* has a second criterion based on v -blocking sets. But the second criterion is weaker than ratification, offering no guarantees to befouled nodes that enjoy quorum intersection.

Now suppose a statement a has the property that no intact node ever votes against it. Then we have no need to accept a and can instead insist that nodes directly ratify a before acting on it. We call such statements *irrefutable*.

Definition (irrefutable). A statement a is *irrefutable* in an FBAS if no intact node can ever vote against it.

Theorem 8 tells us that two intact nodes cannot accept contradictory statements. Thus, while some intact nodes may vote against a statement a that was accepted by an intact node, the statement “an intact node accepted a ” is irrefutable. This suggests holding a second vote to ratify the fact that an intact node accepted a .

Definition (confirm). A quorum U_a in an FBAS *confirms* a statement a iff $\forall v \in U_a, v$ claims to accept a . A node *confirms* a iff it is in such a quorum.

Nodes express that they have accepted statement a by stating “*accept(a)*,” an abbreviation of the statement, “An intact node accepted a .” To confirm a means to ratify *accept(a)*. A well-behaved node v can vote for *accept(a)* only after accepting a , as v cannot assume any particular other nodes are intact. If v itself is befouled, *accept(a)* might be false, in which case voting for it may cost v liveness, but a befouled node has no guarantee of liveness anyway.

The next theorem shows that nodes can rely on confirmed statements without losing optimal safety. Theorem 11 then shows that confirmed statements meet the definition of *agreement* from Section 5.4.2, meaning nodes can rely on confirmed statements without endangering the liveness of intact nodes.

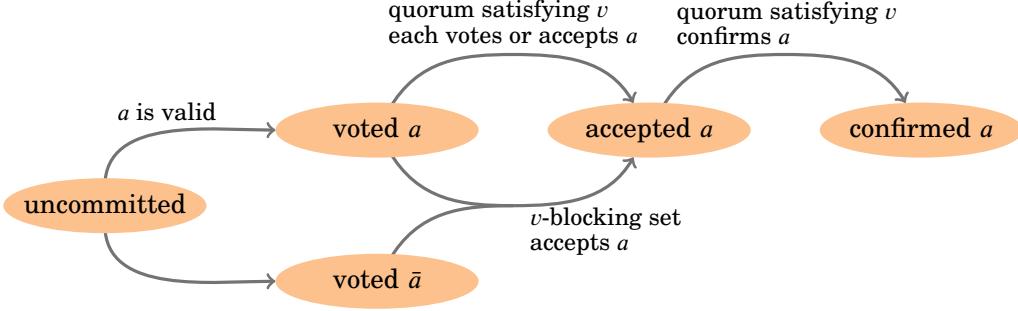
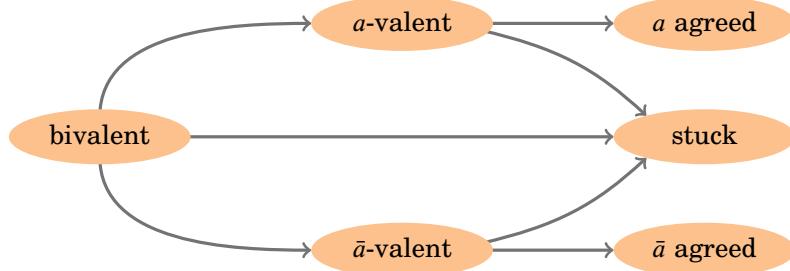
THEOREM 9. *Let $\langle V, Q \rangle$ be an FBAS enjoying quorum intersection despite B , and suppose B contains all ill-behaved nodes. Let v_1 and v_2 be two nodes not in B . Let a and \bar{a} be contradictory statements. If v_1 confirms a , then v_2 cannot confirm \bar{a} .*

PROOF. First note that *accept(a)* contradicts *accept(̄a)*—no well-behaved node can vote for both. Note further that v_1 must ratify *accept(a)* to confirm a . By Theorem 5, v_2 cannot ratify *accept(̄a)* and hence cannot confirm \bar{a} . \square

THEOREM 10. *Let B be the set of befouled nodes in an FBAS $\langle V, Q \rangle$ with quorum intersection. Let U be a quorum containing an intact node ($U \not\subseteq B$), and let S be any set such that $U \subseteq S \subseteq V$. Let $S^+ = S \setminus B$ be the set of intact nodes in S , and let $S^- = (V \setminus S) \setminus B$ be the set of intact nodes not in S . Either $S^- = \emptyset$, or $\exists v \in S^-$ such that S^+ is v -blocking.*

PROOF. If S^+ is v -blocking for some $v \in S^-$, then we are done. Otherwise, we must show $S^- = \emptyset$. If S^+ is not v -blocking for any $v \in S^-$, then, by Theorem 7, either $S^- = \emptyset$ or S^- is a quorum in $\langle V, Q \rangle^B$. In the former case we are done, while in the latter we get a contradiction: By Theorem 1, $U \setminus B$ is a quorum in $\langle V, Q \rangle^B$. Since B is a DSet (by Theorem 3), $\langle V, Q \rangle^B$ must enjoy quorum intersection, meaning $S^- \cap (U \setminus B) \neq \emptyset$. This is impossible, since $(U \setminus B) \subseteq S$ and $S^- \cap S = \emptyset$. \square

THEOREM 11. *If an intact node in an FBAS $\langle V, Q \rangle$ with quorum intersection confirms a statement a , then, whatever subsequently transpires, once sufficient messages are delivered and processed, every intact node will accept and confirm a .*

Fig. 11. Possible states of an accepted statement a at a single node v Fig. 12. Possible system-wide status of a statement a

PROOF. Let B be the DSet of befouled nodes and let $U \not\subseteq B$ be the quorum through which an intact node confirmed a . Let nodes in $U \setminus B$ broadcast $\text{accept}(a)$. By definition, any node v , regardless of how it has voted, accepts a after receiving $\text{accept}(a)$ from a v -blocking set. Hence, these messages may convince additional nodes to accept a . Let these additional nodes in turn broadcast $\text{accept}(a)$ until a point is reached at which, regardless of future communication, no further intact nodes can ever accept a . At this point let S be the set of nodes that claim to accept a (where $U \subseteq S$), let S^+ be the set of intact nodes in S , and let S^- be the set of intact nodes not in S . S^+ cannot be v -blocking for any node in S^- , or else more nodes could come to accept a . By Theorem 10, then, $S^- = \emptyset$, meaning every intact node has accepted a . \square

Figure 11 summarizes the paths an intact node v can take to confirm a . Given no knowledge, v might vote for either a or the contradictory \bar{a} . If v votes for \bar{a} , it cannot later vote for a , but can nonetheless accept a if a v -blocking set accepts it. A subsequent quorum of confirmation messages allows v to confirm a , which by Theorem 11 means the system agrees on a .

5.6. Liveness and neutralization

The main challenge of distributed consensus, whether centralized or not, is that a statement can get stuck in a permanently indeterminate state before the system reaches agreement on it. Hence, a protocol must not attempt to ratify externalized values directly. Should the statement “The value of slot i is x ” get stuck, the system will be forever unable to agree on slot i , losing liveness. The solution is to craft the statements in votes carefully. It must be possible to break a stuck statement’s hold on the question we really care about, namely slot contents. We call the process of obsoleting a stuck statement *neutralization*.

Local state	System-wide status of a
uncommitted	unknown (any)
voted a	unknown (any)
voted \bar{a}	unknown (any)
accepted a	stuck, a -valent, or a agreed
confirmed a	a agreed

Fig. 13. What an intact node knows about the status of statement a

More concretely, Figure 12 depicts the potential status a statement a can have system-wide. Initially, the system is *bivalent*, by which we mean there is one sequence of possible events through which all intact nodes will accept a , and another sequence through which all intact nodes will *reject* a (i.e., accept a statement \bar{a} contradicting a). At some point, one of these two outcomes may cease to be possible. If no intact node can ever reject a , we say the system is *a -valent*; conversely, if no intact node can ever accept a , we say the system is *\bar{a} -valent*.

At the time an FBAS transitions from bivalent to a -valent, there is a possible outcome in which all intact nodes accept a . However, this might not remain the case. Consider a PBFT-like four-node system $\{v_1, \dots, v_4\}$ in which any three nodes constitute a quorum. If v_1 and v_2 vote for a , the system becomes a -valent; no three nodes can ratify a contradictory statement. However, if v_3 and v_4 subsequently vote for \bar{a} contradicting a , it also becomes impossible to ratify a . In this case, a 's state is permanently indeterminate, or *stuck*.

As seen in Figure 10a, even once an intact node accepts a , the system may still fail to reach system-wide agreement on a . However, by Theorem 11, once an intact node confirms a , all intact nodes can eventually come to accept it; hence the system has agreed upon a . Figure 13 summarizes what intact nodes know about the global state of a statement from their own local state.

To preserve the possibility of consensus, a protocol must ensure that every statement is either irrefutable, and hence cannot get stuck, or neutralizable, and hence cannot block progress if stuck. There are two popular approaches to crafting neutralizable statements: the *view-based* approach, pioneered by viewstamped replication [Oki and Liskov 1988] and favored by PBFT [Castro and Liskov 1999]; and the *ballot-based* approach, invented by Paxos [Lamport 1998]. The ballot-based approach may be harder to understand [Ongaro and Ousterhout 2014]. Compounding confusion, people often call viewstamped replication “Paxos” or assert that the two algorithms are the same when they are not [van Renesse et al. 2014].

View-based protocols associate the *slots* in votes with monotonically increasing view numbers. Should consensus get stuck on the i th slot in view n , nodes recover by agreeing that view n had fewer than i meaningful slots and moving to a higher view number. Ballot-based protocols associate the *values* in votes with monotonically increasing ballot numbers. Should a ballot get stuck, nodes retry the same slot with a higher ballot, taking care never to select values that would contradict prior stuck ballots.

This work takes a ballot-based approach, as doing so makes it easier to do away with the notion of a distinguished primary node or leader. For example, leader behavior can be emulated [Lamport 2011b].

6. SCP: A FEDERATED BYZANTINE AGREEMENT PROTOCOL

This section presents the Stellar Consensus Protocol, SCP. At a high level, SCP consists of two sub-protocols: a nomination protocol and a ballot protocol. The nomination

protocol produces candidate values for a slot. If run long enough, it eventually produces the same set of candidate values at every intact node, which means nodes can combine the candidate values in a deterministic way to produce a single composite value for the slot. There are two huge caveats, however. First, nodes have no way of knowing when the nomination protocol has reached the point of convergence. Second, even after convergence, ill-behaved nodes may be able to reset the nomination process a finite number of times.

When nodes guess that the nomination protocol has converged, they execute the ballot protocol, which employs federated voting to commit and abort ballots associated with composite values. When intact nodes agree to commit a ballot, the value associated with the ballot will be externalized for the slot in question. When they agree to abort a ballot, the ballot's value becomes irrelevant. If a ballot gets stuck in a state where one or more intact nodes cannot commit or abort it, then nodes try again with a higher ballot; they associate the new ballot with the same value as the stuck one in case any node believes the stuck ballot was committed. Intuitively, safety results from ensuring that all stuck and committed ballots are associated with the same value. Liveness follows from the fact that a stuck ballot can be neutralized by moving to a higher ballot.

The remainder of this section presents the nomination and ballot protocols. Each is described first in terms of conceptual statements, then as a concrete protocol with messages representing sets of conceptual statements. Finally, Section 6.3 shows the correctness of the protocol. SCP treats each slot completely independently and can be viewed as many separate instances of a single-slot consensus protocol (akin to the “single-decree synod” in Paxos [Lamport 1998]). Concepts such as candidate values and ballots must always be interpreted in the context of a particular slot even if much of the discussion leaves the slot implicit.

6.1. Nomination protocol

Because slots need only be partially ordered, some applications of SCP will have only one plausible ballot per slot. For example, in certificate transparency, each CA may have its own series of slots and sign exactly one certificate tree per slot. However, other applications admit many plausible values per slot, in which case it is helpful to narrow down the possible input values. Our strategy is to begin with a synchronous nomination protocol that achieves consensus under certain timing assumptions, and feed the output of the nomination protocol into an asynchronous ballot protocol whose safety does not depend on timing [Lamport 2011a]. Such an initial synchronous phase is sometimes called a *conciliator* [Aspnes 2010].

The nomination protocol works by converging on a set of candidate values for a slot. Nodes then deterministically combine these candidates into a single *composite* value for the slot. Exactly how to combine values depends on the application. By way of example, the Stellar network uses SCP to choose a set of transactions and a ledger timestamp for each slot. To combine candidate values, Stellar takes the union of their transaction sets and the maximum of their timestamps. (Values with invalid timestamps will not receive enough nominations to become candidates.) Other possible approaches include combining sets by intersection or simply picking the candidate value with the highest hash.

Nodes produce a candidate value x through federated voting on the statement *nominate x*.

Definition (candidate). A node v considers a value x to be a *candidate* when v has confirmed the statement *nominate x*—i.e., v has ratified *accept(nominate x)*.

So long as node v has no candidate values, v may vote in favor of *nominate* x for any value x that passes application-level validity checks (such as timestamps not being in the future). In fact, v should generally re-nominate any values that it sees other nodes nominate, with some rate-limiting discussed below to avoid an explosion of candidates. As soon as v has a candidate value, however, it must cease voting to *nominate* x for any new values x . It should still continue to accept *nominate* statements for new values (when accepted by a v -blocking set) and confirm new *nominate* statements as prescribed by the federated voting procedure.

The nomination protocol enjoys several properties when a system has intact nodes (meaning it has avoided complete failure). Specifically, for each slot:

- (1) Intact nodes can produce at least one candidate value.
- (2) At some point, the set of possible candidate values stops growing.
- (3) If any intact node considers x to be a candidate value, then eventually every intact node will consider x to be a candidate value.

Now consider how the nomination protocol achieves its three properties. Property 1 is achieved because *nominate* statements are irrefutable. Nodes never vote against nominating a particular value, and until the first candidate value is confirmed, intact nodes can vote to nominate any value. So long as any value x passes application-level validity checks, intact nodes can vote for and confirm *nominate* x . Property 2 is ensured because once each intact node confirms at least one candidate value—which will happen in a finite amount of time—no intact nodes will vote to nominate any new values. Hence, the only values that can become candidates are those that already have votes from intact nodes. Property 3 is a direct consequence of Theorem 11.

The nomination process will be more efficient if fewer combinations of values are in play. Hence, we assign nodes a temporary priority and have each node, when possible, nominate the same values as a higher-priority node. More concretely, let H be a cryptographic hash function whose range can be interpreted as a set of integers $\{0, \dots, h_{\max} - 1\}$. (H might be SHA-256 [National Institute of Standards and Technology 2012], in which case $h_{\max} = 2^{256}$.) Let $G_i(m) = H(i, x_{i-1}, m)$ be a slot-specific hash function for slot i , where x_{i-1} is the value chosen for the slot preceding i (or the sorted set of values of all immediate dependencies of slot i when slots are governed by a partial order). Given a slot i and a *round number* n , each node v computes a set of *neighbors* and a *priority* for each neighbor as follows:

$$\begin{aligned} \text{weight}(v, v') &= \frac{\left| \{ q \mid q \in \mathbf{Q}(v) \wedge v' \in q \} \right|}{|\mathbf{Q}(v)|} \\ \text{neighbors}(v, n) &= \{ v' \mid G_i(\mathbf{N}, n, v') < h_{\max} \cdot \text{weight}(v, v') \} \\ \text{priority}(n, v') &= G_i(\mathbf{P}, n, v') \end{aligned}$$

\mathbf{N} and \mathbf{P} are constants to produce two different hash functions. The function $\text{weight}(v, v')$ returns the fraction of slices in $\mathbf{Q}(v)$ containing v' . By using weight as the probability over n that v' appears in $\text{neighbors}(v, n)$, we also reduce the chance that nodes without a lot of trust will dominate a round.

Each node v should initially find a node $v_0 \in \text{neighbors}(v, 0)$ that maximizes $\text{priority}(0, v_0)$ among nodes it can communicate with, then vote to *nominate* the same values as v_0 . Only if $v = v_0$ should v introduce a new value to nominate. v should use timeouts to decide on new *nominate* statements to vote for. After n timeouts, v should

Variable	Meaning
X	The set of values node v has voted to nominate
Y	The set of values node v has accepted as nominated
Z	The set of values that node v considers candidate values
N	The set of the latest NOMINATE message received from each node

Fig. 14. Nomination state maintained by node v for each slotNOMINATE $v \ i \ X \ Y \ D$

This is a message from node v nominating values for slot i . D is v 's quorum slice $\mathbf{Q}(v)$ or a collision-resistant hash of $\mathbf{Q}(v)$. X and Y are from v 's state. The concrete message encodes the following conceptual messages:

- { $\text{nominate } x \mid x \in X$ } (votes to nominate each value in X)
- { $\text{accept}(\text{nominate } x) \mid x \in Y$ } (votes to confirm nominations in Y)

Fig. 15. Message in nomination protocol

find a node $v_n \in \text{neighbors}(v, n)$ maximizing $\text{priority}(n, v_n)$ and vote to nominate everything v_n has voted to nominate.

THEOREM 12. *Eventually, all intact nodes will have the same composite value.*

PROOF. The theorem follows from the three properties of the nomination protocol. Each intact node will only ever vote to nominate a finite number of ballots. In the absence of action by ill-behaved nodes, intact nodes will converge on the same set of candidate values, call it Z . To forestall this convergence, ill-behaved nodes may introduce new candidate values, which for a period may be candidates at some but not all intact nodes. Such values will need to have garnered votes from well-behaved nodes, however, which limits them to a finite set. Eventually, ill-behaved nodes will either stop perturbing the system or run out of new candidate values to inject, in which case intact nodes will converge on Z . \square

6.1.1. Concrete nomination protocol. Figure 14 lists the nomination protocol state a node v must maintain for each slot. X is the set of values x for which v has voted $\text{nominate } x$, Y is the set of values for which v has accepted $\text{nominate } x$, and Z is the set of candidate values—i.e., all values for which a quorum including v has stated $\text{accept}(\text{nominate } x)$. Finally, v maintains N , the latest concrete message from each node. (Technically, X , Y , and Z can all be recomputed from N , but it is convenient to be able to reference them directly.) All four fields are initialized to the empty set. Note that all three of X , Y , and Z are growing over time—nodes never remove a value from these sets.

Figure 15 shows the concrete message that constitutes the nomination protocol. Because X and Y grow monotonically over time, it is possible to determine which of multiple NOMINATE messages from the same node is the latest, independent of network delivery order, so long as D does not change mid-nomination (or D has to be versioned). Only one remote procedure call (RPC) is needed for nomination—the argument is the sender's latest NOMINATE message and the return value is the receiver's. If D or the nominated values are cryptographic hashes, a second RPC should permit retrieval of uncached hash preimages as needed.

Because nodes cannot tell when the nomination protocol is complete anyway, SCP must cope with different composite values at different nodes. As an optimization, then,

nodes can attempt to predict the final composite value before they even have a candidate value. To do this, the composite value can be taken as $\text{combine}(Z)$ when $Z \neq \emptyset$, otherwise $\text{combine}(Y)$ when $Y \neq \emptyset$, otherwise $\text{combine}(X)$ when $X \neq \emptyset$. This means the highest-priority node can optimistically initiate balloting at the same time as nomination, piggybacking its first ballot message PREPARE (described below) on its first NOMINATE message.

6.2. Ballot protocol

Once nodes have a composite value, they engage in the ballot protocol, though nomination may continue to update the composite value in parallel. A ballot b is a pair of the form $b = \langle n, x \rangle$, where $x \neq \perp$ is a value and b is a referendum on externalizing x for the slot in question. The value $n \geq 1$ is a counter to ensure higher ballot numbers are always available. We use C-like notation $b.n$ and $b.x$ to denote the counter and value fields of ballot b , so that $b = \langle b.n, b.x \rangle$. Ballots are totally ordered, with $b.n$ more significant than $b.x$. For convenience, a special invalid null ballot $\mathbf{0} = \langle 0, \perp \rangle$ is less than all other ballots, and a special counter value ∞ is greater than all other counters.

We speak of committing and aborting a ballot b as a shorthand for using federated voting to agree on the statements *commit b* and *abort b*, respectively. For a given ballot, *commit* and *abort* are contradictory, so a well-behaved node may vote for at most one of them. In the notation of Section 5, the opposite of *commit b* would be “ $\overline{\text{commit } b}$,” but *abort b* is a more intuitive notation.

Because at most one value can be chosen for a given slot, all committed and stuck ballots must contain the same value. Roughly speaking, this means *commit* statements are invalid if they conflict with lower-numbered unaborted ballots.

Definition (compatible). Two ballots b_1 and b_2 are *compatible*, written $b_1 \sim b_2$, iff $b_1.x = b_2.x$ and *incompatible*, written $b_1 \not\sim b_2$, iff $b_1.x \neq b_2.x$. We also write $b_1 \lesssim b_2$ or $b_2 \gtrsim b_1$ iff $b_1 \leq b_2$ (or equivalently $b_2 \geq b_1$) and $b_1 \sim b_2$. Similarly, $b_1 \not\lesssim b_2$ or $b_2 \not\gtrsim b_1$ means $b_1 \leq b_2$ (or equivalently $b_2 \geq b_1$) and $b_1 \not\sim b_2$.

Definition (prepared). A ballot b is *prepared* iff every statement in the following set is true: $\{ \text{abort } b_{\text{old}} \mid b_{\text{old}} \not\lesssim b \}$.

More precisely, then, *commit b* is valid to vote for only if b is confirmed prepared, which nodes ensure through federated voting on the corresponding *abort* statements. It is convenient to vote on these statements en masse, so wherever we write “ b is prepared,” the surrounding context applies to the whole set of *abort* statements. In particular, a node votes, accepts, or confirms that b is prepared iff it votes for, accepts, or confirms, respectively, all of these *aborts*.

To commit a ballot b and externalize its value $b.x$, SCP nodes first accept and confirm b is prepared, then accept and confirm *commit b*. Before the first intact node votes for *commit b*, the prepare step, through federated voting, ensures all intact nodes can eventually confirm b is prepared. When an intact node v accepts *commit b*, it means $b.x$ will eventually be chosen. However, as discussed in Section 5.4.1, v must confirm *commit* before acting on it in case v is befouled.

6.2.1. Concrete ballot protocol. Figure 16 illustrates the per-slot state maintained by each node. A node v stores: its current phase φ ; its current ballot b ; the two most recent incompatible ballots it has prepared (p, p'); the lowest (c) and highest (h) ballot, if any, it has voted to *commit* and for which it has not subsequently accepted an *abort* (or for which it has accepted or confirmed a *commit* in later phases); a next value z to try if the current ballot fails; and the latest message received from each node (M). Ballots b , p , p' , and h are non-decreasing within a phase. In addition, if $c \neq \mathbf{0}$ —meaning v may

Variable	Meaning
φ	Current phase: one of PREPARE, CONFIRM, or EXTERNALIZE
b	Current ballot that node v is attempting to prepare and commit ($b \neq \mathbf{0}$)
p', p	The two highest ballots accepted as prepared such that $p' \not\lesssim p$, where $p' = \mathbf{0}$ or $p = p' = \mathbf{0}$ if there are no such ballots
c, h	In PREPARE: h is the highest ballot confirmed as prepared, or $\mathbf{0}$ if none; if $c \neq \mathbf{0}$, then c is lowest and h the highest ballot for which v has voted <i>commit</i> and not accepted <i>abort</i> . In CONFIRM: lowest, highest ballot for which v accepted <i>commit</i> In EXTERNALIZE: lowest, highest ballot for which v confirmed <i>commit</i> Invariant: if $c \neq \mathbf{0}$, then $c \lesssim h \lesssim b$.
z	Value to use in next ballot. If $h = \mathbf{0}$, then z is the composite value (see Section 6.1); otherwise, $z = h.x$.
M	Set of the latest ballot message seen from each node

Fig. 16. Ballot state maintained by each node v for each slotPREPARE $v i b p p' c.n h.n D$

This is a message from node v about slot i . D specifies $\mathbf{Q}(v)$. The other fields reflect v 's state. Values $c.x$ and $h.x$ are elided as $c.x = h.x = b.x$ when $c.n \neq 0$. This concrete message encodes a host of conceptual statements, as follows:

- $\{ \text{abort } b' \vee \text{accept}(\text{abort } b') \mid b' \not\lesssim b \}$ (a vote to prepare b)
- $\{ \text{accept}(\text{abort } b') \mid b' \not\lesssim p \}$ (a vote to confirm p is prepared)
- $\{ \text{accept}(\text{abort } b') \mid b' \not\lesssim p' \}$ (a vote to confirm p' is prepared)
- $\{ \text{commit } b' \mid c.n \neq 0 \wedge c \lesssim b' \lesssim h \}$ (a vote to commit c, \dots, h if $c \neq \mathbf{0}$)

CONFIRM $v i b p.n c.n h.n D$

Sent by v to try to externalize $b.x$ for slot i after accepting a *commit*. Implies $p.x = c.x = h.x = b.x$ in v 's state. For convenience, we also say $p' = \mathbf{0}$ (p' is irrelevant after accepting *commit*). D specifies $\mathbf{Q}(v)$ as above. Encodes:

- Everything implied by PREPARE $v i \langle \infty, b.x \rangle p \mathbf{0} c.n \infty D$
- $\{ \text{accept}(\text{commit } b') \mid c \lesssim b' \lesssim h \}$ (a vote to confirm *commit* c, \dots, h)

EXTERNALIZE $v i x c.n h.n D$

After v confirms *commit* $\langle c.n, x \rangle$ for slot i and externalizes value x , this message helps other nodes externalize x . Implies $c = \langle c.n, x \rangle$ and $h = \langle h.n, x \rangle$. For convenience, we also say $b = p = h = \langle \infty, x \rangle$, and $p' = \mathbf{0}$. Encodes:

- Everything implied by CONFIRM $v i \langle \infty, x \rangle \infty c.n \infty D$
- Everything implied by CONFIRM $v i \langle \infty, x \rangle \infty c.n h.n \{\{v\}\}$

Fig. 17. Messages in SCP's ballot protocol

have participated in ratifying *commit* c —code must ensure $c \lesssim h \lesssim b$. This invariant guarantees a node can always legally vote to prepare its current ballot b .

Figure 17 shows the three ballot protocol messages, with φ determining which one of the three a node can send. Ballot messages may overlap with nomination messages, so that, when $h = \mathbf{0}$, a node may update z in response to a NOMINATE message. Note

that “ $a \vee \text{accept}(a)$ ” is what each node must assert for a quorum to accept a under condition 1 of the definition of accept .

For convenience, when comparing state across nodes, we will identify fields belonging to particular nodes with subscripts. If v is a node, then we write b_v, p_v, p'_v, \dots to denote the values of b, p, p', \dots in node v 's state as described in Figure 16. Similarly, we let v_m denote message m 's sender, and b_m, p_m, p'_m, \dots denote the corresponding values of b, p, p', \dots in v_m 's state as implied by m .

Each node initializes its ballot state for a slot by setting $\varphi \leftarrow \text{PREPARE}$, $z \leftarrow \perp$, $b \leftarrow \langle 0, z \rangle$, $M \leftarrow \emptyset$, and all other fields (p, p', c, h) to the invalid ballot $\mathbf{0}$. While $z = \perp$, a node can receive but not send ballot messages. Once $z \neq \perp$, if $b.n = 0$, a node reinitializes $b \leftarrow \langle 1, z \rangle$ to start sending messages. Nodes then repeatedly exchange messages with peers, sending whichever ballot message is indicated by φ . Upon adding a newly received message m to M_v , a node v updates its state as follows:

- (1) If $\varphi = \text{PREPARE}$ and m lets v accept new ballots as prepared, update p and p' . Afterwards, if either $p \not\geq h$ or $p' \not\geq h$, then set $c \leftarrow \mathbf{0}$.
- (2) If $\varphi = \text{PREPARE}$ and m lets v confirm new higher ballots prepared, then raise h to the highest such ballot and set $z \leftarrow h.x$.
- (3) If $\varphi = \text{PREPARE}$, $c = \mathbf{0}$, $b \leq h$, and neither $p \not\geq h$ nor $p' \not\geq h$, then set c to the lowest ballot satisfying $b \leq c \leq h$.
- (4) If $\varphi = \text{PREPARE}$ and v accepts commit for one or more ballots, set c to the lowest such ballot, then set h to the highest ballot such that v accepts all $\{ \text{commit } b' \mid c \leq b' \leq h \}$, and set $\varphi \leftarrow \text{CONFIRM}$. Also set $z \leftarrow h.x$ after updating h , and unless $h \leq b$, set $b \leftarrow h$.
- (5) If $\varphi = \text{CONFIRM}$ and the received message lets v accept new ballots prepared, raise p to the highest accepted prepared ballot such that $p \sim c$.
- (6) If $\varphi = \text{CONFIRM}$ and v accepts more commit messages or raises b , then let h' be the highest ballot such that v accepts all $\{ \text{commit } b' \mid b \leq b' \leq h' \}$ (if any). If there exists such an h' and $h' > h$, then set $h \leftarrow h'$, and, if necessary, raise c to the lowest ballot such that v accepts all $\{ \text{commit } b' \mid c \leq b' \leq h \}$.
- (7) If $\varphi = \text{CONFIRM}$ and v confirms $\text{commit } c'$ for any c' , set c and h to the lowest and highest such ballots, set $\varphi \leftarrow \text{EXTERNALIZE}$, externalize $c.x$, and terminate.
- (8) If $\varphi \in \{\text{PREPARE}, \text{CONFIRM}\}$ and $b < h$, then set $b \leftarrow h$.
- (9) If $\varphi \in \{\text{PREPARE}, \text{CONFIRM}\}$ and $\exists S \subseteq M_v$ such that the set of senders $\{ v_{m'} \mid m' \in S \}$ is v -blocking and $\forall m' \in S, b_{m'}.n > b_v.n$, then set $b \leftarrow \langle n, z \rangle$, where n is the lowest counter for which no such S exists. Repeat the previous steps after updating b .

While $c = \mathbf{0}$, the above protocol implements federated voting to confirm b is prepared. Once $c \neq \mathbf{0}$, the protocol implements federated voting on $\text{commit } c'$ for every $c \leq c' \leq h$. For the CONFIRM phase, once a well-behaved node accepts $\text{commit } c$, the node never accepts, and hence never attempts to confirm, $\text{commit } c'$ for any $c' \not\sim c$. Once a commit is confirmed, the value of its ballot is safe to externalize assuming quorum intersection.

All messages sent by a particular node are totally ordered by $\langle \varphi, b, p, p', h \rangle$, with φ the most significant and h the least significant field. The values of these fields can be determined from messages, as described in Figure 17. All PREPARE messages precede all CONFIRM messages, which in turn precede the single EXTERNALIZE message for a given slot. The ordering makes it possible to ensure M contains only the latest ballot

from each node without relying on timing to order the messages, since the network may re-order messages.

A few details of the protocol merit explanation. The statements implied by PREPARE of the form “*abort b' v accept(abort b')*” do not specify whether v is voting for or confirming *abort b'*. The distinction is unimportant for the definition of *accept*. Glossing over the distinction allows v to forget about old ballots it voted to *commit* (and hence cannot vote to *abort*), so long as it accepted an *abort* message for them. Indeed, the only time v modifies c when $c \neq \mathbf{0}$ is to set it back to $\mathbf{0}$ after accepting *abort* for every ballot it is voting to *commit* in step 1 on the preceding page. Conversely, the only time v modifies c when $c = \mathbf{0}$ is to set it to a value $c \geq b$ in step 3. Because nodes never vote *abort c* for any $c \geq b$, no past *abort* votes can conflict with *commit c*.

Theorem 11 requires that nodes rebroadcast what they have accepted. It follows from the definition of *prepare* that the two highest incompatible ballots a node has accepted as prepared subsume all ballots the node has accepted as prepared. Hence, including p and p' in every message ensures that nodes converge on h —a confirmed prepared ballot. Note further that the ballots a node *accepts* as prepared must be a superset of the ballots the node *confirms* as prepared; hence, step 2 can never set h such that $h \sim c \neq \mathbf{0}$, as step 1 will set $c \leftarrow \mathbf{0}$ if the new h is incompatible with the old c .

At the time v sends an EXTERNALIZE message, it has accepted $\{\text{commit } b' \mid b' \gtrsim c\}$. More importantly, however, it has confirmed $\{\text{commit } b' \mid c \lesssim b' \lesssim h\}$. v can assert its acceptance of confirmed statements without regard to $Q(v)$, because it has already checked that one of its slices unanimously agrees; this explains the appearance of $\{\{v\}\}$ in place of D for the second implicit CONFIRM message in the description of EXTERNALIZE. Eliminating D allows a single static EXTERNALIZE message to help other nodes catch up arbitrarily far in the future, even if quorum slices have changed significantly in the meantime.

Only one RPC is needed to exchange ballot messages. The argument is the sender's latest message and the return value is the receiver's latest message. As with NOMINATE, if D or the values x in ballots are cryptographic hashes, then a separate RPC is needed to retrieve uncached hash preimages.

6.2.2. Timeouts and ballot updates. If all intact nodes start with the same ballot b , then steps 1 to 9 on the previous page are sufficient to confirm *commit b* and externalize value $b.x$. Unfortunately, if the ballot protocol starts before the nomination protocol has converged, nodes may start off with different values for z . If a ballot fails, or takes long enough that it may fail because of unresponsive nodes, then nodes must time out and try again with a higher ballot. For this reason, nodes employ a timer as follows:

- (a) A node v with $\varphi_v \neq \text{EXTERNALIZE}$ arms a timer whenever $\exists S \subseteq M_v$ such that the set of senders $U = \{v_m \mid m \in S\}$ is a quorum, $v \in U$, and $\forall m \in S, b_m.n \geq b_v.n$.
- (b) If the timer fires, v updates its ballot by setting $b_v \leftarrow \langle b_v.n + 1, z_v \rangle$.

Different nodes may start ballots at different times. However, condition (a) delays setting a timer at a node v that has gotten ahead of a quorum. Conversely, step 9 on the preceding page allows nodes that have fallen too far behind to catch up without waiting for timers. Taken together, these rules ensure that given long enough timers, intact nodes will spend time together on the same ballot; moreover, this time will grow proportionally to the timer duration. To ensure timeouts are long enough without predicting latencies, an implementation can increase the timeout as a function of $b.n$.

6.3. Correctness

An SCP node cannot vote to confirm *commit b* until it has voted to confirm *abort* for all lower-numbered incompatible ballots. Because a well-behaved node cannot accept (and

hence vote to confirm) contradictory statements, this means that for a given $\langle \mathbf{V}, \mathbf{Q} \rangle$, Theorem 5 ensures a set S of well-behaved nodes cannot externalize contradictory values so long as S enjoys quorum intersection despite $\mathbf{V} \setminus S$. This safety holds if \mathbf{V} and \mathbf{Q} change only between slots, but what if they change mid-slot (for instance, in reaction to node crashes)?

To reason about safety under reconfiguration, we join all old and new quorum slice sets, reflecting the fact that nodes may make decisions based on a combination of messages from different configuration eras. To be very conservative, we might require quorum intersection of the aggregation of the present configuration with every past configuration. However, we can relax this slightly by separating nodes that have sent illegal messages from those that have merely crashed.

THEOREM 13. *Let $\langle \mathbf{V}_1, \mathbf{Q}_1 \rangle, \dots, \langle \mathbf{V}_k, \mathbf{Q}_k \rangle$ be the set of configurations an FBAS has experienced during agreement on a single slot. Let $\mathbf{V} = \mathbf{V}_1 \cup \dots \cup \mathbf{V}_k$ and $\mathbf{Q}(v) = \{ q \mid \exists j \text{ such that } v \in \mathbf{V}_j \wedge q \in \mathbf{Q}_j(v) \}$. Let $B \subseteq \mathbf{V}$ be a set such that B contains all ill-behaved nodes that have sent illegal messages, though $\mathbf{V} \setminus B$ may still contain crashed (unresponsive) nodes. Suppose nodes v_1 and v_2 are well-behaved, v_1 externalizes x_1 for the slot, and v_2 externalizes x_2 . If $\langle \mathbf{V}, \mathbf{Q} \rangle^B$ enjoys quorum intersection, then $x_1 = x_2$.*

PROOF. For v_1 to externalize x_1 , it must have ratified $\text{accept}(\text{commit } \langle n_1, x_1 \rangle)$ in collaboration with a pseudo-quorum $U_1 \subseteq \mathbf{V}$. We say pseudo-quorum because U_1 might not be a quorum in $\langle \mathbf{V}_j, \mathbf{Q}_j \rangle$ for any particular j , as ratification may have involved messages spanning multiple configurations. Nonetheless, for ratification to succeed $\forall v \in U_1, \exists j, \exists q \in \mathbf{Q}_j(v)$ such that $q \subseteq U_1$. It follows from the construction of \mathbf{Q} that $q \in \mathbf{Q}(v)$. Hence U_1 is a quorum in $\langle \mathbf{V}, \mathbf{Q} \rangle$. By a similar argument a pseudo-quorum U_2 must have ratified $\text{accept}(\text{commit } \langle n_2, x_2 \rangle)$, and U_2 must be a quorum in $\langle \mathbf{V}, \mathbf{Q} \rangle$. By quorum intersection of $\langle \mathbf{V}, \mathbf{Q} \rangle^B$, there must exist some $v \in \mathbf{V} \setminus B$ such that $v \in U_1 \cap U_2$. By assumption, such a $v \notin B$ could not claim to accept incompatible ballots. Since v confirmed accepting commit for ballots with both x_1 and x_2 , it must be that $x_1 = x_2$. \square

For liveness of a node v , we care about several things when an FBAS has undergone a series of reconfigurations $\langle \mathbf{V}_1, \mathbf{Q}_1 \rangle, \dots, \langle \mathbf{V}_k, \mathbf{Q}_k \rangle$ within a single slot. First, the safety prerequisites of Theorem 13 must hold for v and the set of nodes v cares about, since violating safety undermines liveness and Theorem 11 requires quorum intersection. Second, the set of ill-behaved nodes in the latest state, $\langle \mathbf{V}_k, \mathbf{Q}_k \rangle$, must not be v -blocking, as this could deny v a quorum and prevent it from ratifying statements. Finally, v 's state must never have been poisoned by a v -blocking set falsely claiming to accept a statement.

To summarize, then, if B is the set of nodes that have sent illegal messages, we consider a node v to be *cumulatively intact* when the following conditions hold:

- (1) v is intact in the latest configuration $\langle \mathbf{V}_k, \mathbf{Q}_k \rangle$,
- (2) The aggregation of the present and all past configurations has quorum intersection despite B (i.e., the prerequisite for Theorem 13 holds), and
- (3) B is not v -blocking in $\langle \mathbf{V}_j, \mathbf{Q}_j \rangle$ for any $1 \leq j \leq k$.

The next few theorems show that ill-behaved nodes cannot drive intact nodes into dead-end stuck states:

THEOREM 14. *In an FBAS with quorum intersection, if no intact node is in the EXTERNALIZE phase and an intact node with ballot $\langle n, x \rangle$ arms its timer as described in Section 6.2.2, then, given sufficient communication, every intact node v can set $b_v \geq n$ before any timer fires.*

PROOF. Let $S = \{ v \mid b_v \geq n \}$ be the set of nodes with counters at least n . By assumption, S contains an intact node. Furthermore, because that intact node armed its timer, S must also encompass a quorum. Let S^+ be the intact subset of S , and S^- be the set of intact nodes not in S . By Theorem 10, either $S^- = \emptyset$ (in which case the theorem is trivial), or S^+ is v -blocking for some $v \in S$. By step 9 on page 24, v will adjust its ballot so $b_v.n \geq n$. At this point, repeat the argument with $S \leftarrow S \cup \{v\}$ until such point as $S^- = \emptyset$. \square

THEOREM 15. *Given long enough timeouts, if an intact node has reached the CONFIRM phase with $b.x = x$, then eventually all intact nodes will terminate.*

PROOF. If an intact node has reached the EXTERNALIZE phase, it has confirmed *commit* c for some ballot c . By Theorem 11, all intact nodes will confirm *commit* c , after which they will terminate in step 7 on page 24.

Otherwise, an intact node in the CONFIRM phase has accepted *commit* c where $c = \langle n, x \rangle$. Beforehand, an intact node confirmed c was prepared. By Theorem 11, all intact nodes will eventually have $h \geq c$. Moreover, by Theorem 8, no intact node v can accept *abort* c , so no intact node can accept as prepared any ballot p such that $p \not\geq c$. Hence, after sufficient communication, every intact node will permanently have $h \geq c$. The intact node or nodes with the lowest b will, by Theorem 14, raise their ballots until such point as all intact nodes with armed timers have the same ballot counter. Since they also have identical $z = h.x = x$, they will all have the same ballot. If they cannot complete the protocol because one or more intact nodes have higher ballots, the nodes with higher numbered ballots will not have timers set. Hence, the nodes with lower-numbered ballots will after a timeout set set $b \leftarrow \langle b.n + 1, x \rangle$ until eventually all intact nodes are on the same ballot and can complete the protocol. \square

THEOREM 16. *Regardless of past ill-behavior, given long enough timeouts and periods in which ill-behaved nodes do not send new messages, intact nodes running SCP will terminate.*

PROOF. By Theorem 12, all intact nodes will eventually have identical sets Z of candidate values. Assume this point has passed and every intact node v has the same composite value $z = \text{combine}(Z)$. If no intact node ever confirms any ballot b prepared without $b.x = z$, then after at most one timeout, all new ballots of intact nodes will have value z and, given a sufficient timeout, complete the protocol. By Theorem 15, nodes will also complete if any intact node has progressed beyond the PREPARE phase.

The remaining case is that an intact node has $h \neq \mathbf{0}$ and all intact nodes have $\varphi = \text{PREPARE}$. By Theorem 14, when the intact node or nodes with the highest $b.n$ arm their timers, if timers are long enough, other nodes will catch up. Moreover, by Theorem 11, if timers are long enough, nodes will converge on the value of h (the highest confirmed prepared ballot) before the next timeout, at which point all intact nodes will raise b to the same value and complete the protocol. \square

Theorem 16 assures us there are no dead-end states in SCP. However, a set of ill-behaved nodes with very good timing could perpetually preempt an SCP system by delaying messages so that some fraction of intact nodes update h right before timers fire and the remaining update it after, preventing intact nodes from converging on the next ballot. Nodes can recover from such an attack by removing ill-behaved nodes from their slices.

An alternative would be to add randomness to the protocol, for instance changing step 2 on page 24 to update z with probability $1/2$ (or even with probability proportional to the fraction of the timer remaining). Such an approach would terminate with

probability 1, but in worse expected running time for the common case that most or all nodes are well-behaved or fail-stop.

7. LIMITATIONS

SCP can only guarantee safety when nodes choose adequate quorum slices. Section 3.2 discusses why we can reasonably expect them to do so. Nonetheless, when security depends upon a user-configurable parameter, there is always the possibility people will set it wrong.

Even when people set quorum slices correctly and SCP guarantees safety, safety alone does not rule out other security issues that may arise in a federated system. For example, in a financial market, widely trusted nodes could leverage their position in the network to gain information with which to engage in front running or other unethical activities.

Byzantine nodes may attempt to filter transactions on the input side of SCP while otherwise producing the correct output. If well-behaved nodes accept all transactions, the *combine* function takes the union of transactions, and there are intact nodes, then such filtering will eventually fail to block victim transactions with probability 1, but may nonetheless impose delays.

Though SCP's safety is optimal, its performance and communication latency are not. In the common case that nodes have not previously voted to commit ballots incompatible with the current one, it is possible to reduce the number of communication rounds by one. An earlier version of SCP did so, but the protocol description was more complex. First, it required nodes to cache and retransmit signed messages previously sent by failed nodes. Second, it was no longer possible to gloss over the distinction between votes and confirmations of *abort* statements in PREPARE messages, so nodes had to send around potentially unbounded lists of exceptions to their *abort* votes.

SCP can suffer perpetual preemption as discussed in Section 6.3. An open question is whether, without randomness, a different protocol could guarantee termination assuming bounded communication latency but tolerating Byzantine nodes that continuously inject bad messages at exactly the point where timeouts fire. Such a protocol is not ruled out by the FLP impossibility result [Fischer et al. 1985]. However, the two main techniques to guarantee termination assuming synchrony do not directly apply in the FBA model: PBFT [Castro and Liskov 1999] chooses a leader in round-robin fashion, which is not directly applicable when nodes do not agree on membership. (Possibly something along the lines of priority in Section 6.1 could be adapted.) The Byzantine Generals protocol [Lamport et al. 1982] relays messages so as to compensate for ill-behaved nodes saying different things to different honest nodes, an approach that cannot help when nodes depend on distinct ill-behaved nodes in their slices. Still another possibility might be to leverage both randomness and synchrony to terminate with probability 1, but in shorter expected time than Ben Or-style randomized protocols [Ben-Or 1983] that make no synchrony assumptions. Public coin techniques [?] that speed up randomized centralized Byzantine agreement protocols appear to be difficult to adapt to the federated model, barring some cryptographic breakthrough in federated threshold signatures.

Unfortunately, changing slices mid-slot to accommodate failed nodes is problematic for liveness if a well-behaved node v has ever experienced a wholly malicious and colluding v -blocking set. The good news is that Theorem 13 guarantees safety to any set S of well-behaved nodes enjoying quorum intersection despite $V \setminus S$, even when S has befouled members. The bad news is that updating \mathbf{Q} may be insufficient to unblock nodes if well-behaved nodes were tricked into voting to confirm a bad *commit* message. In such a situation, nodes must disavow past votes, which they can do only by rejoining the system under a new node names. There may exist a way to automate such

recovery, such as having other nodes recognize reincarnated nodes and automatically update their slices.

The FBA model requires continuity of participants over time. Should all nodes simultaneously and permanently leave, restarting consensus would require central coordination or human-level agreement. By contrast, a proof-of-work system such as Bitcoin could undergo sudden complete turnover yet continue to operate with little human intervention. On the other hand, if nodes do return, an FBAS can recover from an arbitrarily long outage, while a proof-of-work scheme would face the possibility of an attacker working on a fork during the outage.

An intriguing possibility is to leverage SCP to mediate tussles [Clark et al. 2005] by voting on changes to configuration parameters or upgrades to an application protocol. One way to do this is to nominate special messages that update parameters. Candidate values could then consist of both a set of values and a set of parameter updates. A big limitation of this approach is that a set of malicious nodes large enough to deny the system a quorum but not large enough to undermine safety could nonetheless trigger configuration changes by lying and putting configuration changes in Y that were never ratified. It remains an open question how to vote on parameter changes in a way that requires the consent of a full quorum but also never jeopardizes liveness.

8. SUMMARY

Byzantine agreement has long enabled distributed systems to achieve consensus with efficiency, standard cryptographic security, and flexibility in designating trusted participants. More recently, Bitcoin introduced the revolutionary notion of decentralized consensus, leading to many new systems and research challenges. This paper introduces federated Byzantine agreement (FBA), a model for achieving decentralized consensus while preserving the traditional benefits of Byzantine agreement. The key distinction between FBA and prior Byzantine agreement systems is that FBA forms quorums from participants' individual trust decisions, allowing an organic growth model similar to that of the Internet. The Stellar Consensus Protocol (SCP) is a construction for FBA that achieves optimal safety against ill-behaved participants.

Acknowledgments

Jed McCaleb inspired this work and provided feedback, terminology suggestions, and help thinking through numerous conjectures. Jessica Collier collaborated on writing the paper. Stan Polu created the first implementation of SCP and provided invaluable corrections, suggestions, simplifications, and feedback in the process. Jelle van den Hoooff provided the key idea to restructure the paper around quorum intersection and federated voting, as well as other crucial suggestions for terminology, organization, and presentation. Nicolas Barry found several bugs in the paper as he implemented the protocol, as well as identifying necessary clarifications. Ken Birman, Bekki Bolt-house, Joseph Bonneau, Mike Hamburg, Graydon Hoare, Joyce Kim, Tim Makarios, Mark Moir, Robert Morris, Lucas Ryan, and Katherine Tom slogged through drafts of the paper, identifying errors and sources of confusion as well as providing helpful suggestions. Eva Gantz provided helpful motivation and references. Winnie Lim provided guidance on figures. The reddit community and Tahoe-LAFS group pointed out a censorship weakness in an earlier version of SCP, leading to the improved nomination protocol. Finally, the author would like to thank the whole Stellar team for their support, feedback, and encouragement.

Disclaimer

Professor Mazières's contribution to this publication was as a paid consultant, and was not part of his Stanford University duties or responsibilities.

REFERENCES

- Eduardo A. Alchieri, Alysson Neves Bessani, Joni Silva Fraga, and Fabíola Greve. 2008. Byzantine Consensus with Unknown Participants. In *Proceedings of the 12th International Conference on Principles of Distributed Systems*. 22–40.
- James Aspnes. 2010. A Modular Approach to Shared-memory Consensus, with Applications to the Probabilistic-write Model. In *Proceedings of the 29th Symposium on Principles of Distributed Computing*. 460–467.
- Rachel Banning-Lover. 2015. Boatfuls of cash: how do you get money into fragile states? (February 2015). <http://www.theguardian.com/global-development-professionals-network/2015/feb/19/boatfuls-of-cash-how-do-you-get-money-into-fragile-states>.
- David Basin, Cas Cremers, Tiffany Hyun-Jin Kim, Adrian Perrig, Ralf Sasse, and Paweł Szalachowski. 2014. ARPKI: Attack Resilient Public-Key Infrastructure. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 382–393.
- Michael Ben-Or. 1983. Another Advantage of Free Choice (Extended Abstract): Completely Asynchronous Agreement Protocols. In *Proceedings of the 2nd Symposium on Principles of Distributed Computing*. 27–30.
- Joseph Bonneau, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A. Kroll, and Edward W. Felten. 2015. Research Perspectives and Challenges for Bitcoin and Cryptocurrencies. In *Proceedings of the 36th IEEE Symposium on Security and Privacy*.
- Gabriel Bracha and Sam Toueg. 1985. Asynchronous Consensus and Broadcast Protocols. *Journal of the ACM* 32, 4 (Oct. 1985), 824–840.
- Danny Bradbury. 2013. Feathercoin hit by massive attack. (June 2013). <http://www.coindesk.com/feathercoin-hit-by-massive-attack/>.
- Vitalik Buterin. 2014. Slasher: A Punitive Proof-of-Stake Algorithm. (January 2014). <https://blog.ethereum.org/2014/01/15/slasher-a-punitive-proof-of-stake-algorithm/>.
- Miguel Castro and Barbara Liskov. 1999. Practical byzantine fault tolerance. In *Proceedings of the 3rd Symposium on Operating Systems Design and Implementation*. 173–186.
- CGAP. 2008. Making Money Transfers Work for Microfinance Institutions. (March 2008). <http://www.cgap.org/sites/default/files/CGAP-Technical-Guide-Making-Money-Transfers-Work-for-Microfinance-Institutions-A-Technical-Guide-to-Developing-and-Delivering-Money-Transfers-Mar-2008.pdf>.
- David D. Clark, John Wroclawski, Karen R. Sollins, and Robert Braden. 2005. Tussle in Cyberspace: Defining Tomorrow's Internet. *IEEE/ACM Transactions on Networking* 13, 3 (June 2005), 462–475.
- crazyearner. 2013. TERRACOIN ATTACK OVER 1.2TH ATTACK CONFIRMD [sic]. (July 2013). <https://bitcointalk.org/index.php?topic=261986.0>.
- Kourosh Davarpanah, Dan Kaufman, and Ophelie Pubellier. 2015. NeuCoin: the First Secure, Cost-efficient and Decentralized Cryptocurrency. (March 2015). <http://www.neucoin.org/en/whitepaper/download>.
- Asli Demirguc-Kunt, Leora Klapper, Dorothe Singer, and Peter Van Oudheusden. 2015. *The Global Findex Database 2014 Measuring Financial Inclusion Around the World*. Policy Research Working Paper 7255. World Bank. http://www-wds.worldbank.org/external/default/WDSContentServer/WDSP/IB/2015/04/15/090224b082dca3aa/1_0/Rendered/PDF/TheGlobalFinion0around0the0world.pdf.
- John R. Douceur. 2002. The Sybil Attack. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems*. 251–260.
- Cynthia Dwork, Nancy Lynch, and Larry Stockmeyer. 1988. Consensus in the Presence of Partial Synchrony. *Journal of the ACM* 35, 2 (April 1988), 288–323.
- Cynthia Dwork and Moni Naor. 1992. Pricing via Processing or Combatting Junk Mail. In *Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*. 139–147.
- Ittay Eyal and Emin Gün Sirer. 2013. Majority is not Enough: Bitcoin Mining is Vulnerable. (November 2013). <http://arxiv.org/abs/1311.0243>.
- Michael J. Fischer, Nancy A. Lynch, and Michael S. Paterson. 1985. Impossibility of Distributed Consensus with One Faulty Process. *Journal of the ACM* 32, 2 (April 1985), 374–382.
- Ghassan O. Karame, Elli Androulaki, and Srdjan Capkun. 2012. Double-spending fast payments in bitcoin. In *Proceedings of the 2012 ACM conference on Computer and communications security*. 906–917.
- Tiffany Hyun-Jin Kim, Lin-Shung Huang, Adrian Perrig, Collin Jackson, and Virgil Gligor. 2013. Accountable Key Infrastructure (AKI): A Proposal for a Public-key Validation Infrastructure. In *Proceedings of the 22nd International Conference on World Wide Web*. 679–690.

- Sunny King and Scott Nadal. 2012. PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake. (August 2012). <http://peercoin.net/assets/paper/peercoin-paper.pdf>.
- Jae Kwon. 2014. Tendermint: Consensus without Mining. (2014). <http://tendermint.com/docs/tendermint.pdf>.
- Leslie Lamport. 1998. The Part-Time Parliament. 16, 2 (May 1998), 133–169.
- Leslie Lamport. 2011a. Brief Announcement: Leaderless Byzantine Paxos. In *Proceedings of the 25th International Conference on Distributed Computing*. 141–142.
- Leslie Lamport. 2011b. Byzantizing Paxos by Refinement. In *Proceedings of the 25th International Conference on Distributed Computing*. 211–224.
- Leslie Lamport, Robert Shostak, and Marshall Pease. 1982. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems* 4, 3 (July 1982), 382–401.
- Adam Langley. 2015. Maintaining digital certificate security. (March 2015). <http://googleonlinesecurity.blogspot.com/2015/03/maintaining-digital-certificate-security.html>.
- Ben Laurie, Adam Langley, and Emilia Kasper. 2013. *Certificate Transparency*. RFC 6962. Internet Engineering Task Force (IETF). <http://tools.ietf.org/html/rfc6962>.
- Jinyuan Li and David Mazières. 2007. Beyond One-third Faulty Replicas in Byzantine Fault Tolerant Systems. In *Proceedings of the 4th Symposium on Networked Systems Design and Implementation*. 131–144.
- Marcela S. Melara, Aaron Blankstein, Joseph Bonneau, Michael J. Freedman, and Edward W. Felten. 2014. CONIKS: A Privacy-Preserving Consistent Key Service for Secure End-to-End Communication. Cryptology ePrint Archive, Report 2014/1004. (December 2014). <http://eprint.iacr.org/2014/1004>.
- Microsoft. 2013. Fraudulent Digital Certificates Could Allow Spoofing. Microsoft Security Advisory 2798897. (January 2013). <https://technet.microsoft.com/en-us/library/security/2798897.aspx>.
- Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. (2008). <http://bitcoin.org/bitcoin.pdf>.
- National Institute of Standards and Technology. 2012. *Secure Hash Standard (SHS)*. Federal Information Processing Standards Publication 180-4. <http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf>.
- William B. Norton. 2010. The Art of Peering: The Peering Playbook. (August 2010). <http://drpeering.net/white-papers/Art-Of-Peering-The-Peering-Playbook.html>.
- Karl J. O'Dwyer and David Malone. 2014. Bitcoin Mining and its Energy Footprint. In *Irish Signals and Systems Conference*. Limerick, Ireland, 280–285.
- Brian M. Oki and Barbara H. Liskov. 1988. Viewstamped Replication: A New Primary Copy Method to Support Highly-Available Distributed Systems. In *Proceedings of the 7th Symposium on Principles of Distributed Computing*. 8–17.
- Diego Ongaro and John Ousterhout. 2014. In Search of an Understandable Consensus Algorithm. In *2014 USENIX Annual Technical Conference*. 305–319.
- Marshall Pease, Robert Shostak, and Leslie Lamport. 1980. Reaching Agreement in the Presence of Faults. *Journal of the ACM* 27, 2 (April 1980), 228–234.
- Claire Provost. 2013. Why do Africans pay the most to send money home? (January 2013). <http://www.theguardian.com/global-development/2013/jan/30/africans-pay-most-send-money>.
- David Schwartz, Noah Youngs, and Arthur Britto. 2014. The Ripple Protocol Consensus Algorithm. (2014). https://ripple.com/files/ripple_consensus_whitepaper.pdf.
- Dale Skeen and Michael Stonebraker. 1983. A Formal Model of Crash Recovery in a Distributed System. *IEEE Transactions on Software Engineering* 9, 3 (May 1983), 219–228.
- Robbert van Renesse, Nicolas Schiper, and Fred B. Schneider. 2014. Vive la Différence: Paxos vs. Viewstamped Replication vs. Zab. *IEEE Transactions on Dependable and Secure Computing* (September 2014).

A. GLOSSARY OF NOTATION

Notation	Name	Definition
iff		An abbreviation of “if and only if”
$f : A \rightarrow B$	function	Function f maps each element of set A to a result in set B .
$f(x)$	application	The result of calculating function f on argument x
\bar{a}	complement	An overbar connotes the opposite, i.e., \bar{a} is the opposite of a .
$\langle a_1, \dots, a_n \rangle$	tuple	A structure (compound value) with field values a_1, \dots, a_n
$A \wedge B$	logical and	Both A and B are true.
$A \vee B$	logical or	At least one, possibly both, of A and B are true.
$\exists e, C(e)$	there exists	There is at least one value e for which condition $C(e)$ is true.
$\forall e, C(e)$	for all	$C(e)$ is true of every value e .
$\{a, b, \dots\}$	set	A set containing the listed elements (a, b, \dots)
$\{e \mid C(e)\}$	set-builder	The set of all elements e for which $C(e)$ is true
\emptyset	empty set	The set containing no elements
$ S $	cardinality	The number of elements in set S
$e \in S$	element of	Element e is a member of set S .
$A \subseteq B$	subset	Every member of set A is also a member of set B .
$A \subsetneq B$	strict subset	$A \subseteq B$ and $A \neq B$.
2^A	powerset	The set of sets containing every possible combination of members of A , i.e., $2^A = \{B \mid B \subseteq A\}$
$A \cup B$	union	The set containing all elements that are members of A or members of B , i.e., $A \cup B = \{e \mid e \in A \vee e \in B\}$
$A \cap B$	intersection	The set containing all elements that are members of both A and B , i.e., $A \cap B = \{e \mid e \in A \wedge e \in B\}$
$A \setminus B$	set difference	The set containing every element of A that is not a member of B , i.e., $A \setminus B = \{e \mid e \in A \wedge e \notin B\}$
/	not	Negates a symbol’s meaning. E.g., $e \notin A$ means $e \in A$ is false, while $\nexists e, C(e)$ means no e exists such that $C(e)$ is true.



Tether: Fiat currencies on the Bitcoin blockchain

Abstract. A digital token backed by fiat currency provides individuals and organizations with a robust and decentralized method of exchanging value while using a familiar accounting unit. The innovation of blockchains is an auditable and cryptographically secured global ledger. Asset-backed token issuers and other market participants can take advantage of blockchain technology, along with embedded consensus systems, to transact in familiar, less volatile currencies and assets. In order to maintain accountability and to ensure stability in exchange price, we propose a method to maintain a one-to-one reserve ratio between a cryptocurrency token, called tethers, and its associated real-world asset, fiat currency. This method uses the Bitcoin blockchain, Proof of Reserves, and other audit methods to prove that issued tokens are fully backed and reserved at all times.

Table of Contents

[Table of Contents](#)

[Introduction](#)

[Technology Stack and Processes](#)

[Tether Technology Stack](#)

[Flow of Funds Process](#)

[Proof of Reserves Process](#)

[Implementation Weaknesses](#)

[Main Applications](#)

[For Exchanges](#)

[For Individuals](#)

[For Merchants](#)

[Future Innovations](#)

[Multi-sig and Smart Contracts](#)

[Proof of Solvency Innovations](#)

[Conclusion](#)

[Appendix](#)

[Audit Flaws: Exchanges and Wallets](#)

[Limitations of Existing Fiat-pegging Systems](#)

[Market Risk Examples](#)

[Legal and Compliance](#)

[Glossary of Terms](#)

[References](#)

Introduction

There exists a vast array of assets in the world which people freely choose as a store-of-value, a transactional medium, or an investment. We believe the Bitcoin blockchain is a better technology for transacting, storing, and accounting for these assets. Most estimates measure global wealth around 250 trillion dollars [1] with much of that being held by banks or similar financial institutions. The migration of these assets onto the Bitcoin blockchain represents a proportionally large opportunity.

Bitcoin was created as “an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party.”[2]. Bitcoin created a new class of digital currency, a decentralized digital currency or cryptocurrency¹.

Some of the primary advantages of cryptocurrencies are: low transaction costs, international borderless transferability and convertibility, trustless ownership and exchange, pseudo-anonymity, real-time transparency, and immunity from legacy banking system problems [3]. Common explanations for the current limited mainstream use of cryptocurrencies include: volatile price swings, inadequate mass-market understanding of the technology, and insufficient ease-of-use for non-technical users.

The idea for asset-pegged cryptocurrencies was initially popularized² in the Bitcoin community by the Mastercoin white paper authored by J.R. Willett in January 2012[4]. Today, we’re starting to see these ideas built with the likes of BitAssets, Ripple, Omni, Nxt, NuShares/Bits, and others. One should note that all Bitcoin exchanges and wallets (like Coinbase, Bitfinex, and Coinapult) which allow you to hold value as a fiat currency already provide a *similar* service in that users can avoid the volatility (or other traits) of a particular cryptocurrency by selling them for fiat currency, gold, or another asset. Further, almost all types of existing financial institutions, payment providers, etc, which allow you to hold fiat value (or other assets) subsequently provide a similar service. In this white paper we focus on applications wherein the fiat value is stored and transmitted with software that is open-source, cryptographically secure, and uses distributed ledger technology, i.e. a true cryptocurrency.

While the goal of any successful cryptocurrency is to completely eliminate the requirement of trust, each of the aforementioned implementations either rely on a trusted third party or have other technical, market-based, or process-based drawbacks and limitations³.

¹ For definitions throughout, see [Glossary of Terms](#)

² But has been discussed since Dr. Szabo’s proposed BitGold [5]

³ Summarized in the Appendix, here: [Limitations of Existing Fiat-pegging Systems](#)

In our solution, fiat-pegged cryptocurrencies are called “tethers”. All tethers will initially⁴ be issued on the Bitcoin blockchain via the Omni Layer protocol and so they exist as a cryptocurrency token. Each tether unit issued into circulation is backed in a one-to-one ratio (i.e. one Tether USDT is one US dollar) by the corresponding fiat currency unit held in deposit by Hong Kong based Tether Limited. Tethers may be redeemable/exchangeable for the underlying fiat currency pursuant to Tether Limited’s terms of service or, if the holder prefers, the equivalent spot value in Bitcoin. Once a tether has been issued, it can be transferred, stored, spent, etc just like bitcoins or any other cryptocurrency. The fiat currency on reserve has gained the properties of a cryptocurrency and its price is permanently *tethered* to the price of the fiat currency.

Our implementation has the following advantages over other fiat-pegged cryptocurrencies:

- Tethers exist on the Bitcoin blockchain rather than a less developed/tested “altcoin” blockchain nor within closed-source software running on centralized, private databases.
- Tethers can be used just like bitcoins, i.e. in a p2p, pseudo-anonymous, decentralized, cryptographically secure environment.
- Tethers can be integrated with merchants, exchanges, and wallets just as easily as Bitcoin or any other cryptocurrencies can be integrated.
- Tethers inherit the properties of the Omni Layer protocol which include: a decentralized exchange; browser-based, open-source, wallet encryption; Bitcoin-based transparency, accountability, multi-party security and reporting functions.
- Tether Limited employs a simple but effective approach for conducting Proof of Reserves which significantly reduces our counterparty risk as the custodian of the reserve assets.
- Tether issuance or redemption will not face any pricing or liquidity constraints. Users can buy or sell as many tethers as they want, quickly, and with very low fees.
- Tethers will not face any market risks⁵ such as Black Swan events, liquidity crunches, etc as reserves are maintained in a one-to-one ratio rather than relying on market forces.
- Tether’s one-to-one backing implementation is easier for non-technical users to understand as opposed to collateralization techniques or derivative strategies.

At any given time the balance of fiat currency held in our reserves will be equal to (or greater than) the number of tethers in circulation. This simple configuration most easily supports a reliable Proof of Reserves process; a process which is fundamental to maintaining the price-parity between tethers in circulation and the underlying fiat currency held in reserves. In this paper we provide evidence⁶ that shows exchange and

⁴ More Bitcoin 2.0 protocols will come soon, like Ripple, Nxt, etc

⁵ See Appendix, section: [Market Risk Examples](#)

⁶ See section: [Proof of Solvency Process](#)

wallet audits (in their current state) are very unreliable (i.e. flaws in Proof of Solvency[6] methods) and instead propose that exchanges and wallets *outsource* the custody of user funds to us via tethers.

Users can purchase tethers from Tether.to (our web-wallet) or from supported exchanges such as Bitfinex who support tethers as a deposit and withdrawal method. Users can also transact and store tethers with any Omni Layer enabled wallet like Ambisafe, Holy Transaction or Omni Wallet. Other exchanges, wallets, and merchants are encouraged to reach out to us about integrating tether as a surrogate for traditional fiat payment methods.

We recognize that our implementation isn't perfectly decentralized⁷ since Tether Limited must act as a centralized custodian of reserve assets (albeit tethers in circulation exist as a decentralized digital currency). However, we believe this implementation sets the foundation for building future innovations that will eliminate these weaknesses, create a robust platform for new products and services, and support the growth and utility of the Bitcoin blockchain over the long run. Some of these innovations include:

- Mobile payment facilitation between users and other parties, including other users and merchants
- Instant or near-instant fiat value transfer between decentralized parties (such as multiple exchanges)
- Introduction to the use of smart contracts and multi-signature capabilities to further improve the general security process, Proof of Reserves, and enable new features.

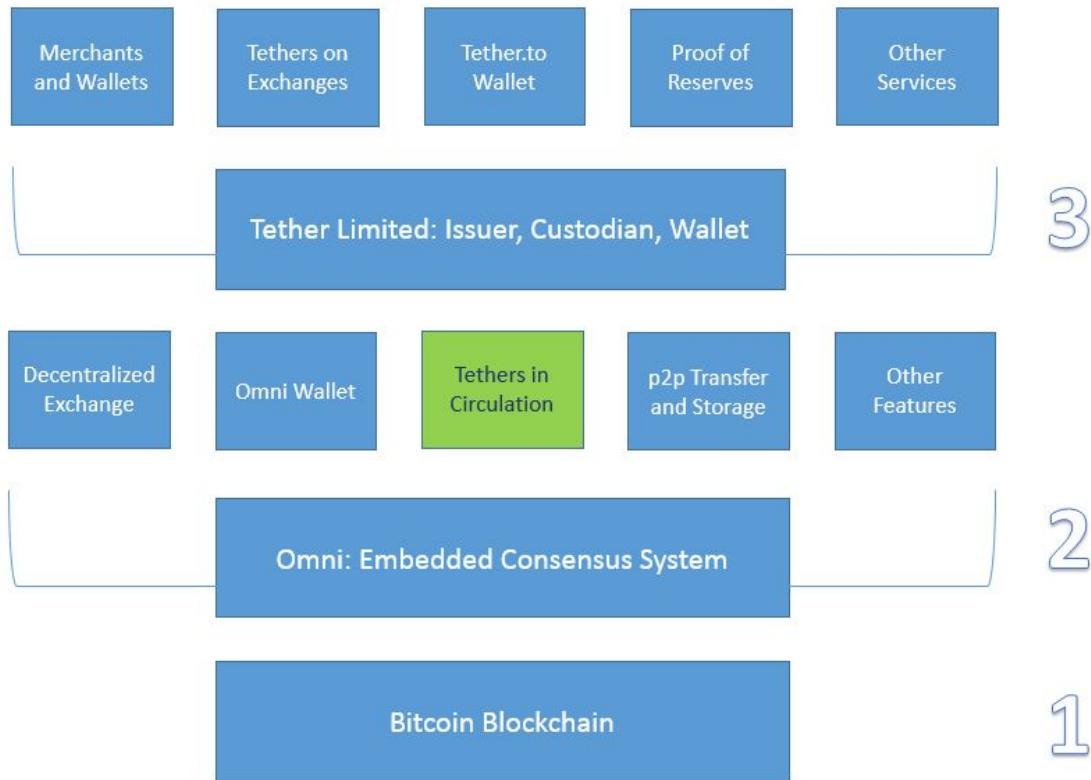
Technology Stack and Processes

Each tether issued into circulation will be backed in a one-to-one ratio with the equivalent amount of corresponding fiat currency held in reserves by Hong Kong based Tether Limited. As the custodian of the backing asset we are acting as a trusted third party responsible for that asset. This risk is mitigated by a simple implementation that collectively reduces the complexity of conducting both fiat and crypto audits while increasing the security, provability, and transparency of these audits.

Tether Technology Stack

The stack has 3 layers, and numerous features, best understood via a diagram

⁷ See section: [Implementation Weaknesses](#)



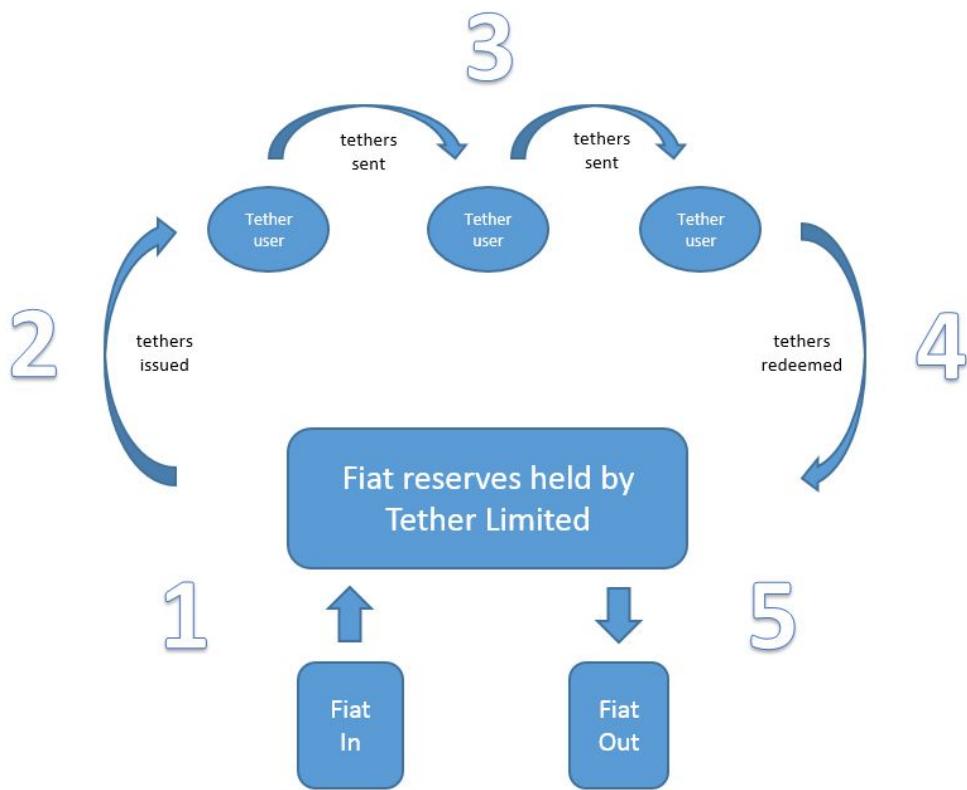
Here is a review of each layer.

- 1) The first layer is the Bitcoin blockchain. The Tether transactional ledger is embedded in the Bitcoin blockchain as meta-data via the embedded consensus system, Omni.
- 2) The second layer is the Omni Layer protocol. Omni is a foundational technology that can:
 - a) Grant (create) and revoke (destroy) digital tokens represented as meta-data embedded in the Bitcoin blockchain; in this case, fiat-pegged digital tokens, tethers.
 - b) Track and report the circulation of tethers via Omnichest.info (Omni asset ID #31, for example, represents TetherUSD) and Omnicore API.
 - c) Enable users to transact and store tethers and other assets/tokens in a:
 - i) p2p, pseudo-anonymous, cryptographically secure environment.
 - ii) open-source, browser-based, encrypted web-wallet: Omni Wallet.
 - iii) multi-signature and offline cold storage-supporting system
- 3) The third layer is Tether Limited, our business entity primarily responsible for:
 - a) Accepting fiat deposits and issuing the corresponding tethers
 - b) Sending fiat withdrawals and revoking the corresponding tethers
 - c) Custody of the fiat reserves that back all tethers in circulation

- d) Publicly reporting Proof of Reserves and other audit results
- e) Initiating and managing integrations with existing Bitcoin/blockchain wallets, exchanges, and merchants
- f) Operating Tether.to, a web-wallet which allows users to send, receive, store, and convert tethers conveniently.

Flow of Funds Process

There are five steps in the lifecycle of a tether, best understood via a diagram.



Step 1 - User deposits fiat currency into Tether Limited's bank account.

Step 2 - Tether Limited generates and credits the user's tether account. Tethers enter circulation. Amount of fiat currency deposited by user = amount of tethers issued to user (i.e. 10k USD deposited = 10k tetherUSD issued).

Step 3 - Users transact with tethers⁸. The user can transfer, exchange, and store tethers via a p2p open-source, pseudo-anonymous, Bitcoin-based platform.

Step 4 - The user deposits tethers with Tether Limited for redemption into fiat currency.

Step 5 - Tether Limited destroys the tethers and sends fiat currency to the user's bank account.

Users can obtain tethers outside of the aforementioned process via an exchange or another individual. Once a tether enters circulation it can be traded freely between any business or individual. For example, users can purchase tethers from Bitfinex, with more exchanges to follow soon.

The main concept to be conveyed by the Flow of Funds diagram is that Tether Limited is the only party who can issue tethers into circulation (create them) or take them out of circulation (destroy them). This is the main process by which the system solvency is maintained.

Proof of Reserves Process

Proof of Solvency, Proof of Reserves, Real-Time Transparency, and other similar phrases have been growing and resonating across the cryptocurrency industry.

Exchange and wallets audits, in their current form, are very unreliable. Insolvency has occurred numerous times in the Bitcoin ecosystem, either via hacks, mismanagement, or outright fraud. Users must be diligent with their exchange selection and vigilant in their use of exchanges. Even then, a savvy user will not be able to fully eliminate the risks. Further, there are exchange users like traders and businesses who must keep non-trivial fiat balances in exchanges at all times. In financial language, this is known as the “counterparty risk” of storing value with a third party.

We believe it's safe to conclude that exchange and wallet audits in their current form are not very reliable. These processes do not guarantee users that a custodian or exchange is solvent. Although there have been great contributions to improving the exchange audit processes, like the Merkle tree approach[6], major flaws⁹ still remain.

Tether's Proof of Reserves configuration is novel because it simplifies the process of proving that the total number of tethers in circulation (liabilities) are always fully backed by an equal amount of fiat currency held

⁸ See benefits of using tethers in the section: [Main Applications](#)

⁹ See section: [Audit Flaws: Exchanges and Wallets](#)

in reserve (assets). In our configuration, each tetherUSD in circulation represents one US dollar held in our reserves (i.e. a one-to-one ratio) which means the system is fully reserved when the sum of all tethers in existence (at any point in time) is exactly equal to the balance of USD held in our reserve. Since tethers live on the Bitcoin blockchain, the provability and accounting of tethers at any given point in time is trivial. Conversely, the corresponding total amount of USD held in our reserves is proved by publishing the bank balance and undergoing periodic audits by professionals. Find this implementation further detailed below:

- Tether Limited issues all tethers via the Omni Layer protocol. Omni operates on top of the Bitcoin blockchain and therefore all issued, redeemed, and existing tethers, including transactional history, are publicly auditable via the tools provided at Omnichest.info.
 - The Omnichest.info asset ID for tetherUSD is #31.
 - Here is a link: <http://omnichest.info/lookupsp.aspx?sp=31>
 - Let the total number of tethers issued under this asset ID be denoted as TUSDissue
 - Let the total number of tethers redeemed under this asset ID be denoted as TUSDredeem
 - Let the total number of tethers in circulation at any time be denoted as TUSD
 - $TUSD = TUSDissue - TUSDredeem$
 - TUSD = “Total Property Tokens” @ <http://omnichest.info/lookupsp.aspx?sp=31>
- Tether Limited has a bank account which will receive and send fiat currency to users who purchase/redeem tethers directly with us.
 - Let the total amount deposited into this account be denoted as DUSDdepo
 - Let the total amount withdrawn from this account be denoted as DUSDwithd
 - Let the dollar balance of this bank account be denoted as DUSD
 - $DUSD = DUSDdepo - DUSDwithd$
- Each tether issued will be backed by the equivalent amount of currency unit (one tetherUSD equals one dollar). By combining the above crypto and fiat accounting processes, we conclude the “Solvency Equation” for the Tether System.
 - The Solvency Equation is simply $TUSD = DUSD$.
 - Every tether issued or redeemed, as publicly recorded by the Bitcoin blockchain will correspond to a deposit or withdrawal of funds from the bank account.
 - The provability of TUSD relies on the Bitcoin blockchain as discussed previously.
 - The provability of DUSD will rely on several processes:
 - We publish the bank account balance on our website’s Transparency page.
 - Professional auditors will regularly verify, sign, and publish our underlying bank balance and financial transfer statement.

Users will be able to view this information from our Transparency Page, which will look like:



For clarity, we'd like to acknowledge that the Tether System¹⁰ is different than the Tether.to web-wallet in terms of Proof of Reserves. In this paper, we mostly focus on Proof of Reserves for the Tether System; i.e. all tethers in circulation at any point in time. The Tether.to wallet is a consumer facing web-wallet operating on closed-source code and centralized servers. Conducting a Proof of Reserves for this wallet is fundamentally different than what we've outlined for the Tether System.

We're planning the deployment of a PoR-based transparency solution for the Tether.to wallet. We believe it will be the most advanced PoR system in existence today. It overcomes almost all of the challenges outlined in the appendix¹¹ on this topic. Mind you, users can always secure tethers through managing the private keys themselves or through Omni Wallet.

Implementation Weaknesses

We understand that our implementation doesn't immediately create a fully trustless cryptocurrency system. Mainly because users must trust Tether Limited and our corresponding legacy banking institution to be the custodian of the reserve assets. However, almost all exchanges and wallets (assuming they hold USD/fiats) are subject to the same weaknesses. Users of these services are already subject to these risks. Here is a summary of the weaknesses in our approach:

- We could go bankrupt
- Our bank could go insolvent
- Our bank could freeze or confiscate the funds
- We could abscond with the reserve funds

¹⁰ See [Glossary of Terms](#)

¹¹ See [Audit Flaws: Exchanges and Wallets](#)

- Re-centralized of risk to a single point of failure

Observe that almost all digital currency exchanges and wallets (assuming they hold USD/flat) already face many of these challenges. Therefore, users of these services are already subject to these risks. Below we describe how each of these concerns are being addressed.

We could go bankrupt - In this case, the business entity Tether Limited would go bankrupt but client funds would be safe, and subsequently, all tethers will remain redeemable. Most security breaches on Bitcoin businesses have targeted cryptocurrencies rather than bank accounts. Since all tethers exist on the Bitcoin blockchain they can be stored by individuals directly through securing their own private keys.

Our bank could go insolvent - This is a risk faced by all users of the legacy financial system and by all exchange operators. Tether Limited currently has accounts with Cathay United Bank and Hwatai Bank in Taiwan, both of whom are aware and confident that Tether's business model is acceptable. Additional banking partners are being established in other jurisdictions to further mitigate this concern.

Our bank could freeze or confiscate the funds - Our banks are aware of the nature of Bitcoin and are accepting of Bitcoin businesses. They also provide banking services to some of the largest Bitcoin exchanges globally. The KYC/AML processes we follow are also used by the other digital currency exchanges they currently bank. They have assured us we are in full compliance¹².

We could abscond with the reserve assets - The corporate charter is public¹³ as well as the business owners names, locations, and reputations. Ownership of the account is legally bound to the corporate charter. Any transfers in or out of the bank account will have the associated traces and are bound by rigid internal policies.

Re-centralization of risk to a single point of failure - We have some ideas on how to overcome this and we'll be sharing them in upcoming blog and product updates. There are many ways to tackle this problem. For now, this initial implementation gets us on the right track to realize these innovations in following versions. By leveraging the platforms we have chosen, we have reduced the centralization risk to one singular responsibility: the creation and redemption of tokens. All other aspects of the system are decentralized.

¹² See section on [Legal and Compliance](#) for more information

¹³ Same as footnote #10

Main Applications

In this section we'll summarize and discuss the main applications of tethers across the Bitcoin/blockchain ecosystem and for other consumers globally. We break up the beneficiaries into three user groups: Exchanges, Individuals, and Merchants.

The main benefits, applicable to all groups:

- Properties of Bitcoin bestowed upon other asset classes
- Less volatile, familiar unit of account
- World's assets migrate to the Bitcoin blockchain

For Exchanges

Exchange operators understand that accepting fiat deposits and withdrawals using legacy financial systems can be complicated, risky, slow, and expensive. Some of these issues include:

- Identifying the right payment providers for your exchange
 - irreversible transactions, fraud protection, lowest fees, etc
- Integrating the platform with banks who have no APIs
- Liaising with these banks to coordinate compliance, security, and to build trust
- Prohibitive costs for small value transfers
- 3-7 days for international wire transfers to clear
- Poor and unfavorable currency conversion fees

By offering tethers, an exchange can relieve themselves of the above complications and gain additional benefits, such as:

- Accept crypto-fiat as deposit/withdrawal/storage method rather than using a legacy bank or payment provider
 - Allows users to move fiat in and out of exchange more freely, quickly, cheaply
- Outsource fiat custodial risk to Tether Limited - just manage cryptos
- Easily add other tethered fiat currencies as trading pairs to the platform
- Secure customer assets purely through accepted crypto-processes
 - Multi-signature security, cold and hot wallets, HD wallets, etc

- Conduct audits easier and more securely in a purely crypto environment
- Anything one can do with Bitcoin as an exchange can be done with tethers

Exchange users know how risky it can be to hold fiat currencies on an exchange. With the growing number of insolvency events it can be quite dangerous. As mentioned previously, we believe that using tethers exposes exchange users to less counterparty risk than continually holding fiat on exchanges. Additionally, there are other benefits to holding tethers, explained in the next section.

For Individuals

There are many types of individual Bitcoin users in the world today. From traders looking to earn profits daily; to long term investors looking to store their Bitcoins securely; to tech-savvy shoppers looking to avoid credit card fees or maintain their privacy; to philosophical users looking to change the world; to those looking to remit payments globally more effectively; to those in third world countries looking for access to financial services for the first time; to developers looking to create new technologies; to all those who have found many uses for Bitcoin. For each of these individuals, we believe tethers are useful in similar ways, like:

- Transact in USD/fiat value, pseudo-anonymously, without any middlemen/intermediaries
- Cold store USD/fiat value by securing one's own private keys
- Avoid the risk of storing fiat on exchanges - move crypto-fiat in and out of exchanges easily
- Avoid having to open a fiat bank account to store fiat value
- Easily enhance applications that work with bitcoin to also support tether
- Anything one can do with Bitcoin as an individual one can also do with tether

For Merchants

Merchants want to focus on their business, not on payments. The lack of global, inexpensive, ubiquitous payment solutions continue to plague merchants around the world both large and small. Merchants deserve more. Here are some of the ways tether can help them:

- Price goods in USD/fiat value rather than Bitcoin (no moving conversion rates/purchase windows)
- Avoid conversion from Bitcoin to USD/fiat and associated fees and processes
- Prevent chargebacks, reduce fees, and gain greater privacy
- Provide novel services because of fiat-crypto features
 - Micropayments, gift cards, more
- Anything one can do with Bitcoin as a merchant one can also do with tether

Future Innovations

Multi-sig and Smart Contracts

Proof of Solvency Innovations

Conclusion

Tether constitutes the first Bitcoin-based fiat-pegged cryptocurrencies in existence today. Tether is based on the Bitcoin blockchain, the most secure and well-tested blockchain and public ledger in existence. Tethers are fully reserved in a one-to-one ratio, completely independent of market forces, pricing, or liquidity constraints. Tether has a simple and reliable Proof of Reserves implementation and undergoes regular professional audits. Our underlying banking relationships, compliance, and legal structure provide a secure foundation for us to be the custodian of reserve assets and issuer of tethers. Our team is composed of experienced and respected entrepreneurs from the Bitcoin ecosystem and beyond.

We are focused on arranging integrations with existing businesses in the cryptocurrency space. Business like exchanges, wallets, merchants, and others. We're already integrated with Bitfinex, HolyTransaction, Omni Wallet, Poloniex, C-CEX, and more to come. Please reach out to us to find out more.

Appendix

Audit Flaws: Exchanges and Wallets

Here is a summary of the current flaws found in technology-based¹⁴ exchange and wallet audits.

In the Merkle tree[6] approach users must manually report that their balances (user's leaf) have been correctly incorporated in the liability declaration of the exchange (the Merkle hash of the exchange's database of user balances). This proposed solution works if enough users verify that their account was included in the tree, and in a case where their account is not included this instance would be reported. One potential risk is that an exchange database owner could produce a hash that is not the true representation of

¹⁴ As opposed to hiring a professional auditor

the database at all; it hashes an incomplete database which would reduce its apparent liabilities to customers, making them appear solvent to a verifying party. Here are some scenarios where a fraudulent exchange would exclude accounts and :

- “Bitdust” Accounts: Inactive or low activity accounts would lower the chance that an uninterested user would check or report inconsistencies. In some cases these long-tail accounts could represent a significant percentage of the exchange’s liabilities.
- “Colluding Whales” Attack: There is evidence that large Bitcoin traders are operating on various exchanges and moving markets significantly. Such traders need to have capital reserves at the largest exchanges to quickly execute orders. Often, traders choose exchanges that they “trust”. In this way they can be assured that should a hack or liquidity issue arise, they have priority to get their money out. In this case, the exchange and trader could collude to remove the whales account balance from the database before it’s hashed.
- Key Rental Attack: To pass the audit, a malicious exchange could rent the private keys to bitcoins they do not own. This would make them appear solvent by increasing their assets without any acknowledgment that those funds were loaned to them. Likewise, they could “borrow” fiat currency to do the same.
- There are more attacks not discussed here.

Reaching Statistical Significance (reporting completeness): Even outside of these three attack vectors, a database that has been manipulated may never be detected if a sufficient number of users are not validating balances. The probability of getting 100% of the users to verify balances is likely zero, even with proper incentivization structure for users to verify their balances. Therefore, auditors would need statistical tools to make statements about the validity of an exchange’s database based on sampling frequency, size, and other properties.

Currently users have no way to receive compensation by legal means in case something goes wrong with the exchange. For example, when Mt.Gox closed operations, many users might not have independently recorded their account balances (prints screens, signed messages to themselves, etc) in a way that could conclusively prove to law enforcement that this exchange’s I.O.U’s actually existed. Such users are at the mercy of the exchange to somehow publish a record of that hash tree or original database.

The proposed structure in which these audits would be performed still contains some subtle but important flaws. In particular, the data reporting (hash tree) on the institution’s website gives no guarantee at all to

users, as a malicious exchange could publish different states/balances to different groups of users, or retroactively change the state. Thus it is fundamental to publish this data through a secure broadcast channel, e.g. the Bitcoin blockchain.

Privacy is a barrier to entry for the adoption of an automated/open auditing system. While some progress has been made towards better privacy there is no perfect solution yet. Further, to build up an accurate user verified liability space, these users will have to report account balances with the exchange and Bitcoin addresses. Some users likely would not report this information regardless of the incentive, therefore providing cryptographically secure privacy whilst obtaining the reporting goal is paramount.

Time Series: the Merkle tree hash is a single snapshot of the database at a single point in time. Not having a somewhat continuous time series of the database opens significant attack vectors. Additionally, a time series of user reported information would also be required for piecing together the history of any reported incidents of fraud.

Trusted Third Parties: All of the current exchange audits have relied on some “reputable” trusted third party to make some type of verification. In the Coinbase audit [7], that was Andreas Antonopoulos, in the Kraken audit [8], that was Stefan Thomas. If we absolutely must rely on a trusted third party then some audit standards and procedures should ensure this weakness is fortified.

Limitations of Existing Fiat-pegging Systems

Here's a list of some of the common drawbacks and limitations of existing fiat-pegging systems.

- The systems are based on closed-source software, running on private, centralized databases, fundamentally no different than Paypal or any other existing mass-market retail/institutional asset trading/transfer/storage system.
- Decentralized systems that rely on altcoin blockchains which haven't been stress-tested, developed, or reviewed as closely as other blockchains, like Bitcoin.
- Pegging processes that rely on hedging derivative meta-assets, efficient market theory, or collateralization of the underlying asset, wherein liquidity, transferability, security, and other issues can exist.

- Lack of transparency and audits for the custodian, either crypto, fiat, or relating to their own internal ledgers (same as closed source and centralised databases).
- Reliance on legacy banking systems and trusted third parties (bank account owners) as a transfer and settlement mechanism for reserve assets.

Market Risk Examples

In the collateralization method, market risk exists because the price of the asset being used as collateral can move in an adverse direction to the price of the asset it's backing/pegging. This would cause the total value of the collateral to become less than the total value of the issued asset and make the system insolvent. This risk is mitigated by the custodian closing the position before this happens; that is, when the collateral price equals the pegged asset price then the collateral is liquidated (sold on the open market) and the position is closed. A great approach, with merit, and used in many liquid markets across the traditional banking and financial markets. However, as we saw from the global financial crisis, situations can arise in which the acceleration of such events causes a "liquidity crunch" and thus the collateral is unable to be liquidated fast enough to meet trading obligations, subsequently creating losses. With the cryptocurrency markets being so small and volatile, this type of event is much more likely. Additionally, the overall approach suffers from other liquidity and pricing constraints since there must be a sufficient supply of users posting collateral for the creation of the pegged-assets to exist in the first place.

In the derivatives approach, the price of the asset is pegged through entering one of several derivatives strategies, such as: swap strategies, covered and naked options strategies, various futures and forwards strategies. Each strategy has their own strengths and weaknesses, the discussion of which we won't engage in here. To summarize, each of these pegging processes themselves have similar "market risk" characteristics as the aforementioned collateralization method. It should be noted that the two methods are not mutually exclusive and often paired in a specific trading, hedging, or risk management function at legacy system financial institutions.

Finally, understand that we believe some combination of the above approaches may become a secure, reliable, and generally risk-free process for backing/pegging assets; however, at this point in time, this is not a direction we feel is feasible to take to ensure liquidity and price stability. Further, we believe that a reserve-based approach will always be in existence and complement these other approaches as the entire industry grows. As advances in technology continue, we will evaluate and incorporate any benefits available while maintaining the guarantee of 100% redeemability.

Legal and Compliance

Tether Limited (“Tether”) is a limited company incorporated pursuant to the Hong Kong Companies Ordinance. It is wholly owned by Tether Holdings Limited, a BVI business company incorporated pursuant to the BVI Business Companies Act, 2004.

Tether is registered as a Money Services Business with the Financial Crimes Enforcement Network of the U.S. Department of the Treasury (MSB Registration Number 31000058542968). Tether is establishing a relationship with a U.S. financial institution for purposes of better servicing Tether users in the United States.

Tether is concluding a principal–agency agreement with RenRenBee Limited (“RenRenBee”). RenRenBee is licensed as a Money Services Operator by the Hong Kong Customs and Excise Department (Licence No. 13-09-01265). Pursuant to the agreement, RenRenBee will provide anti-money laundering compliance work and customer due diligence procedures as agent for Tether as principal.

Through these and other measures, Tether is undertaking customer due diligence, record-keeping, and reporting procedures consistent with U.S. law and with the Hong Kong Anti-Money Laundering and Counter-Terrorist Financing (Financial Institutions) Ordinance.

Tether Limited currently has accounts with Cathay Bank and Hwatai Bank in Taiwan, both of whom are aware and confident that Tether’s business model is acceptable.

These banks are satisfied with our processes and also satisfied that our business operates in accordance with Taiwan off-shore banking regulations, as all of the banks had been requested to check this with their own legal, compliance and head-office before opening accounts (also at our own request). It was our goal from the beginning to have a compliant operation and to provide the maximum level of comfort to our banking partners here. In addition these banks have and are working with other Bitcoin based businesses.

Glossary of Terms

Digital currency: As defined by http://en.wikipedia.org/wiki/Digital_currency

Cryptocurrency or decentralized digital currency: any type of cryptocurrency that is open-source, cryptographically secure, and uses a distributed ledger. See: <http://en.wikipedia.org/wiki/Cryptocurrency>

Real-world currency, or fiat currency, or national/sovereign currency: all types of currency that are not cryptocurrencies as defined above.

Cryptocurrency system: A collection of software and processes primarily created to enable the existence of a cryptocurrency.

Legacy financial system: any financial system that is not a cryptocurrency system.

Utility-backed digital tokens, a.k.a Dapps: A decentralized digital token whose value is derived from the usefulness of its application rather than just being a value transfer system.

Asset-backed/pegged cryptocurrency: Any cryptocurrency whose price is pegged to a real-world asset, i.e. its not a “utility-backed” cryptocurrency.

Tether(s): a single unit (or multiple units) of fiat-pegged cryptocurrency issued by Tether Limited

TetherUSD or tUSD: a single unit of crypto-USD issued by Tether Limited

TUSD: collective amount of tUSD in circulation at any point in time.

Tether System: collectively refers to all process and technologies that enable tethers to exist

Proof of Reserves: The process by which the issuer of any asset-backed decentralized digital token, cryptographically/mathematically proves that all tokens that have been issued are fully reserved and backed by the underlying asset.

References

- [1] <https://www.thefinancialist.com/wp-content/uploads/2012/10/2012-GlobalWealthReport-.pdf>
- [2] <https://bitcoin.org/bitcoin.pdf>
- [3] http://www.deloitte.com/assets/Dcom-UnitedStates/Local%20Assets/Documents/FSI/us_fsi_BitcointheNewGoldRush_031814.pdf
- [4] <https://github.com/mastercoin-MSC/spec>
- [5] <http://unenumerated.blogspot.com/2005/12/bit-gold.html>
- [6] <https://iwilcox.me.uk/2014/proving-bitcoin-reserves>
- [7] <http://antonopoulos.com/2014/02/25/coinbase-review/>
- [8] <http://www.coindesk.com/krakens-audit-proves-holds-100-bitcoins-reserve/>



VECHAIN DEVELOPMENT PLAN

THIS IS NOT A WHITE PAPER !

No, it is NOT!

Preface

The VeChain team and the VeChain Blockchain platform has been running for more than two years.

Fortunately, while running down the path of Blockchain which everyone has big hopes on right now, we met many people sharing the same goal and lots of enterprise customers that dare to explore this new area. We also met many passionate business partners and co-workers with strong believes. Moreover, we accumulated lots of experience of business use cases for different industries and we kept adjusting and making corrections during the process so that we can continue on searching the right way to use this “secret technology” that may change the world.

Our original vision has never been changed. The dream is still same as before,

Building a trust-free and distributed business ecosystem to enable transparent information flow, efficient collaboration, and high speed value transferring.

Table of Content

1.	Concept Background	5
1.1	Call for ICO	5
1.2	The Understanding of Blockchain Technology	6
1.2.1	Synergy and Value Transfer	6
1.2.2	Data and Information Symmetry	7
1.3	VeChain's Vision.....	8
1.3.1	Distributed Business Ecosystem.....	9
1.3.2	The "Blood" in the Distributed Ecosystem – VeChain Token(VET).....	10
1.4	VeChain's Attitude on Blockchain Technology	12
2.	Methodology and Technical Support.....	13
2.1	Methodology	13
2.2	Technique Support.....	14
2.3	Technical Structure	15
2.4	Achieve the Technical Details.....	18
2.4.1	VeChain ID Creation and Hashing.....	18
2.4.2	Storage of VID on Blockchain.....	19
2.4.3	Digital Ownership on Blockchain	20
2.4.4	Data Hashed Storage (proof of data)	21
2.4.5	API Gateway.....	22
2.4.6	Service Discovery (SDP).....	23
2.4.7	Micro-Service	24
2.4.8	Hashed Storage Service (HSS)	24
2.5	Blockchain and IoT	26
2.5.1	The Issue of IoT	26
2.5.2	Blockchain and IoT	26
2.5.3	VeChain and IoT	28
2.6	Technical Testing	29
2.7	Technology Development's Path and Plan.....	31
3.	The Industrial Application and Expansion.....	34
3.1	Fashion and Luxury Industry	35
3.2	Food Safety	36
3.3	Car Industry	39
3.4	Supply Chain Industry	40
3.5	The Agricultural Industry.....	41
3.6	Blockchain Government Affairs	42
3.7	This is just the Beginning	44
4.	Governance Structure and Management Philosophy	45

4.1 The Establishment of VeChain Foundation.....	46
4.2. Governance Principle	46
4.3 VeChain Governance Model	48
4.3.1 Strategic Steering Committee.....	49
4.3.2 General Secretary	50
4.3.3 Technical Audit Committee.....	50
4.3.4 Remuneration and Nomination Committee	50
4.3.5 Public Relation Committee.....	51
4.3.6 Supervisory Committee	51
4.3.7 Other Functional Department	51
4.4 VeChain Human Resource Management	51
4.5 Risk Assessment and Decision Making Mechanism of VeChain Foundation	52
4.6 VeChain Foundation Economy.....	52
4.6.1 Funding Sources	53
4.6.2 Fund Budgeting	54
4.6.3 Fund Use Restriction	56
4.6.4 Financial Planning and Implementation Reports.....	57
4.6.5 Digital Asset Management	57
4.7 Legal Compliance Matters and Other Matters	57
5. Introduction of the Team and Team Member.....	58

1. Concept Background

1.1 Call for ICO

The Blockchain technology is experiencing rapid development especially in recent one or two years. The change is so fast for us entrepreneur. Despite the direction is technology development and expansion, or application research, we face changes all the time. Every time we site together, discuss and conclude "the only thing that does not change is change itself", just like a poet. From the second half of 2015, with an article from The Economist "Blockchain: The Trust Machine", the Blockchain technology started to walk out of the geek community and rapidly gained worldwide attention in various industries.

The term "Blockchain" is no longer an obscure technical term for many people. Lots of the new ideas and projects are coming out, which includes many imaginative models and new directions for Blockchain. At the end of last year, the Blockchain technology was even written directly into the national "13th Five-Year plan", which encouraged us many peers. In addition, it attracts so many aspiring young people to join the Blockchain industry.

Needless to say, Blockchain has been recognized by the world as a new generation of powerful technology. Blockchain is considered as being able to change the world again just like what the Internet technology did. In addition, based on the pattern of human technology development, the development process of the Blockchain technology will suddenly get speeds up by a huge margin without doubt. We believe that it will have substantial breakthroughs and extensive expansion in the upcoming few years for the Blockchain industry.

However, the reality is tougher than expected. The application direction of Blockchain is either for financial industry or non-financial industry. For financial industry, it is so obvious with its high standard of compliance, so it is very hard to make a break through. For non-financial industry, it has a variety of collaborative cooperation modes but all these participants lack of drive to move further. That is why even though there is some new concepts derived from Blockchain, only few practical Blockchain business application has been established. Even when some project get partially established can make the teams feel excited.

Although everything is hard at the beginning, there is always someone who is very careful and willing to be the first one to get into the field. In order to reduce the possibility of failure as much as possible, we want to share VeChain with the investors, enterprise customers, cooperative partners and colleagues – a product that we started making strategic plan two years ago. It has been through several platform software updates, many practical cases and debates with arguments at

so many sleepless nights. Finally we created some matured ideas and we want to CALL for the ICO for VeChain project.

1.2 The Understanding of Blockchain Technology

1.2.1 Synergy and Value Transfer

In the world of traditional business, different varieties of collaborative and business operations as well as the whole financial industry, which is at the top of the "food chain", shows trust is the biggest cost in the field. Though Blockchain carries a "trust aura". The Blockchain technology is widely accepted around the world since the famous article "Blockchain: the trust machine" published by the Economist.

The essence of a Blockchain is an Internet protocol and a collection of technologies about Trust. We can define the meaning of Blockchain from three dimensions - data, system and application:

- *From the data point of view: Blockchain is a distributed database system that is continuously updated in chronological order. The data can only be added but not tampered with.*
- *From the system point of view: Blockchain is a distributed deployment and real-time synchronization system, allows participants from different parties to create and maintain the data through mechanism for consensus. It makes each active node on the Blockchain has exactly the same data.*
- *From the application point of view: Blockchain is a standard global platform allows multiple participants to connect at the same time and records all digital objects, users, and their relative operations on this platform.*

With the development of information technology and Internet, the application of various systems makes the collaboration more convenient and efficient. Because of the existence of trust issue, the majority of such efficient collaboration exists mainly within an enterprise or a certain organization. However, people are using the methods and tools from 40 years ago when it comes to the collaboration between different enterprises. The majority of collaboration is still completed by e-mail. System interfacing is actually not as simple as imagined. Since it involves data security, trade secrets, cooperation, trust and other problems. The connection is not just a technical issue. In addition, due to the same problem, the financial service that match and support all kinds of business needs improvement in both efficiency and cost.

For example, a classic business collaborative mode includes supply chain (as the graphic shows below), brand, manufacturer, distributed retailer, consumer and regulator. All the parties share the same goal: to achieve the same value of

improving the life quality of the consumer. However, even if the different enterprises worked together for the same goal, due to the lack of sufficient trust guarantee, the cooperation is still on a peer-to-peer manner and with traditional communication tools, and the data exchange will be very inefficient and expensive. In such a traditional product life cycle, even if the logistics could be relatively smooth and efficient, the flow of information is often fragmented and the transfer of funds is also relatively slow. For the participants on the whole supply chain, the utilization rate of funds has always been quite a headache.

The Blockchain technology can help us to establish a new trust-free sharing business collaboration model (as the graphic shows below). Various parties can ensure the security of data in a more convenient and smoother manner. With the support of a more timely and accurate information flow, the value transmission in the ecological environment can be developed and executed during the business activities. This way each enterprise can increase the utilization rate of funds, and greatly improve the speed of value transmission in order to support more business development.

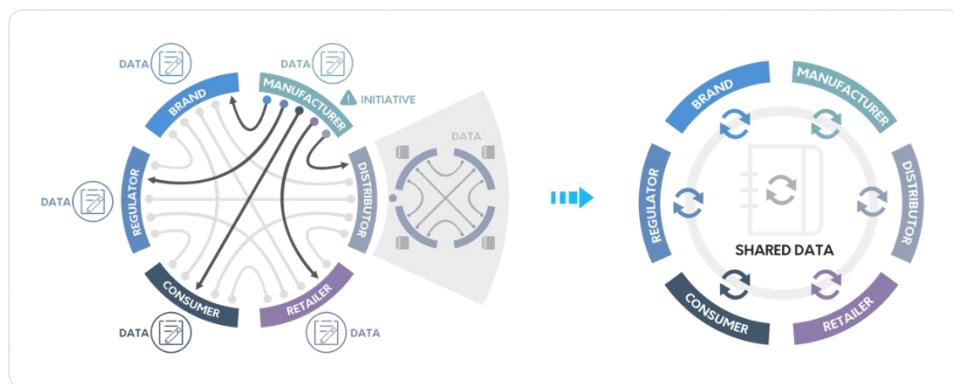


Figure 1.2.1 Distributed business collaboration from traditional business collaboration to Blockchain

1.2.2 Data and Information Symmetry

Most enterprises have three types of data:

- 1) Public data, such as enterprise data that is publicly available on the official website.
- 2) Private data, such as enterprise product developing documents, financial reports for non-listed companies.
- 3) Permitted shared data, usually exists between different cooperation partners. Such as the record data from the identification, logistics, payment information and after-sale service for different level enterprises.

The first and second kind of data are generally well understood. The interesting part is the third type of data, which is usually converted into private data by the participants.

For example, after a car is sold, maintenance data exists in the 4S store or the maintenance business. When the owners need to buy insurance, the insurance company spend huge cost to buy those maintenance data as a service provider. Or, based on the market demand, this could become as a new data service. This service can collect data, centralize maintenance and management, and get paid by providing data to participants who are interested in those data. For users, the risk comes from the centralized/ isolated data, where stay in a variety of online automotive service platform. This kind of service breaks the asymmetric information (due to the difference of region and time) by using information technology, and constructs a new centralized asymmetric information, and then produce profit base on this.

We BELIEVE that the Blockchain technology can continuously break those asymmetric information and allow data to return to the real owner eventually. For instance, in the car scenario described above, the data generated by the owner when he used the car should naturally be owned by the owner. That means the data generated from car maintenance should belong to car owner without doubt. While enjoying other service like insurance later, insurance company can reduce data audit costs through user authorized credible data. Thus car owners could enjoy premiums service with lower cost.

The Blockchain technology allows data owners really own the data, and let the data owner have the authority to choose whether to share their data, which completely breaks the traditional asymmetric of the centralized information. In this way, the value goes back to where they were. The data owned by one side needs to be shared and maintained by all parties. Thus new values are generated and is properly allocated in the activities of multi-party participation.

1.3 VeChain's Vision

What does the VeChain want to do? **The vision of VeChain is to build a trust-free and distributed business ecosystem based on the Blockchain technology self-circulated and expanding.**

- In this ecosystem, the information is relatively **transparent** and symmetrical. A large portion of the source of the profit comes from the realization of true value, and only a small portion of it comes from asymmetric information (absolute symmetry does not exist).
- In this ecosystem, each business party can reduce the **potential trust issue** between different parties. This makes business cooperation simpler, more efficient, low cost, and the business can concentrate resources on more advanced technology, better product and service to create more value.
- In this ecosystem, each person and each enterprise can find their own

place. Based on their contribution and value, they can obtain relatively fair reward.

- In this ecosystem, the technology of Blockchain should have room for all aspects of business, including commercial activities and economic activities that should be supported.
- In this ecosystem, the value is in a **closed loop that keep expanding** accompanied by the development of commercial activities with high-speed transmission. The form of value may be commodities, services, or direct fund.

1.3.1 Distributed Business Ecosystem

In the ecological environment made by VeChain, there are several main types of participants:

1) Enterprise organization

All kinds of enterprise organization that provide products and services to end-users to meet all the needs such as various manufacturing enterprises, brands, service providers for end-users, and so on.

2) Application service provider

It means an enterprise that provides various application development and services for enterprise organizations and users on VeChain Blockchain. The product or service can be a variety of decentralized applications and services to users, technical products and related services for all enterprises and institutions, functions of the government agencies, regulators and third party credit service providers.

For example, end-user oriented Internet platforms like BAT, sharing products and service providers like Uber, Didi, Airbnb;

For instance, enterprise oriented technology, product, service provider such as Oracle, IBM. Supply chain services providers of commodity enterprises, third party credit service providers such as PwC, DNV, GL and financial service providers such as banks and insurance companies.

3) Smart contract service provider

Organizations provide VeChain smart contract technology service to enterprises, and allow the end- enterprise or service providers to develop Blockchain applications in a faster and more convenient manner.

4) VeChain network node provider

Enterprises and organizations that directly participate in the Blockchain network and maintain a certain number of nodes to protect the overall network security.

Maintaining a specific function node to provide related services, such as customs, quality inspection node, audit node, wallet service, and user private key management service provider.

5) VeChain Foundation

VeChain is responsible for the construction of Blockchain network, technology research and development, upgrade and maintenance and other basic technical services. Meanwhile, in the initial stage it is responsible for business development, creating reference cases, encouraging and supporting more of the new smart contract service providers and existing technology enterprises to transform. Based on the demands of ecological development, it provides support to more technology companies on offering Blockchain services, such as wallet development, payment services, private key management, internal exchange, smart contract templates etc.

6) End-user

The service target of end-enterprise, end-users and service provider (investors) enjoy the bonus from the future commercial ecology development together.

These participants set up the whole VeChain distributed business ecosystem. On one hand, it can form an effective closed loop. On the other hand, it can connect and assimilate with the environment outside of the ecosystem and constantly grow itself, as figure shows below:

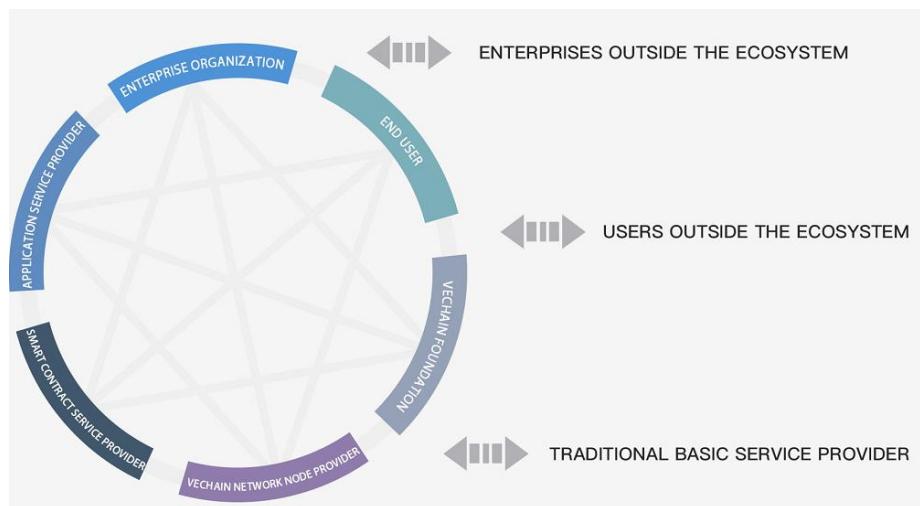


Figure 1.3.1 Distributed business ecosystem environment

1.3.2 The "Blood" in the Distributed Ecosystem – VeChain Token(VET)

If the entire distributed business ecosystem is a body, then the Blockchain infrastructure is the skeleton and various application services are the muscles and organs. Such body needs the circulation of the blood, and the blood is the VeChain tokens - VeChain Token(VET), carrying the value transfer function of the entire Blockchain network and various commercial activities running on it. VeChain Token(VET) will be openly available for sale in a variety of ways in this ICO.

As the carrier of value transmission in the whole ecosystem, VeChain Token(VET) flows through the smart contract which describes and executes the cooperation among the parties, as well as forming a special closed loop with an open interface. On one hand, the value transfer very fast in the ecological. On the other hand, it opens and communicates with outside the ecosystem as a medium, and further expanding the scope of ecology.

The main function of VeChain Token(VET) is to circulate as much as possible and to let each participant use it. So we will sell more than **70%** of the total amount of tokens to communities, businesses, and users.

Just as the graphic shows above:

- 1) This loop begins with the end-user and enterprises as investors to use ETH to obtain VeChain Token(VET) in the beginning. VeChain uses ETH to perform technical development, commercial application cooperation promotion, and Blockchain services support.
- 2) VeChain Foundation receives VeChain Token(VET) from each smart contract development and service provider to pay for the GAS needed to run the smart contract and maintain the operation of each business smart contract. The 75% to 99% of the VeChain Token(VET) income will be awarded as a node reward to the node provider, while the remaining 25% will be used for the daily operation, business promotion and technical development of the VeChain Foundation.
- 3) Smart contract service providers use VeChain Token(VET) to pay for GAS and provide smart contract service of BaaS (Blockchain as a service). According to different business rules and contributions, each participant receives VeChain Token(VET) from its client - Application Development provider provides smart contract services through collecting VeChain Token(VET).
- 4) Application provider develops and processes based on end-user's needs with the foundation of smart contract service, as well as providing products for the application of the traditional enterprise customers or end-users and receive VeChain Token(VET) as corporate income.
- 5) End-users can pay VeChain Token(VET) to obtain enterprise products and services.

Of course, such ecological development will experience different stages, and maintain an open status. A better fusion with the traditional business world will help the transformation of traditional commercial enterprises, and then expand such a distributed business ecosystem.

In this process, there must be a variety of new technology service enterprises, to provide a bridge for communication and value transfer between traditional

commercials and VeChain's distributed business environment.

VeChain will be responsible for the actual development. On the other hand, we will encourage and support outstanding teams to join us so we can have a better understand to the various sectors of all enterprises. We can apply better and more focus on developing and make the right people to do what they do best.

According to the actual experience of the past two years, we have summed up a few reference methods to carry out this ecological promotion:

- 1) The breakthrough point should select the enterprise with most "Blockchain" strategy. These enterprises value greatly on the development of Blockchain technology in the future.
- 2) Initial cases should be combined with the real enterprise issue, which can solve the actual problems, or may bring new value.
- 3) The business scenario has the multi participants, and space for deeper expansion.
- 4) Target enterprises, target cases in the industry or in different industry has considerable influence.

In this ecological development, the tactics needs to expand both horizontally and vertically:

- 1) Horizontally, make more duplicated expansion of the same types of enterprises within the same industry.
- 2) Vertically, the expansion of different enterprises and participants.

More participants will bring more extensive collaboration, more efficient value flow, give birth to a new and strong coupling business model, and then build a future distributed business ecology.

1.4 VeChain's Attitude on Blockchain Technology

The development of any new technology is bound to go through several important stages:

- ✓ The first stage, **technical barriers stage**; at this stage, being capable or not capable plays a very big difference; doing well and not well is not very obvious.
- ✓ The second stage is the **business barrier stage**. At this stage, the development of technology has been advancing by leaps and bounds, and with the trend of social resources, more and more talents have been pouring in. More technical theories and skills are being shared, and technical barriers are becoming increasingly blurred. To do or not to do already is not a problem; well done and badly becomes prominent. The key point of this stage is whether we can apply the technology skillfully and reasonably to the actual commercial products and services and

- produce greater value.
- ✓ The third stage is the **scale barrier stage**. At this stage, the snowball effect is very obvious, and the scale advantage is becoming more and more important. More business activities and social activities focus on one or more ecological environments, and the more participants, the faster they develop.
- ✓ The fourth stage is **the subdividing the vertical phase**. At this stage, the industry scale and pattern are basically formed, and new breakthroughs are made. The new breakthrough comes from the division of vertical areas with more concentrated resources advantages to produce better products, services and values.
- ✓ The fifth stage, **the birth of the new technological revolution**. More advanced technology was born in the human pursuit of higher value, and then enter the next cycle.

Blockchain has no exemption to this route. Although the Blockchain technology itself still has a long way to go, there is lots of space for improvement. Nevertheless, as things stand now, we have unwittingly entered the **early second stage** of Blockchain.

So, this *non-whitepaper* does not include mysterious algorithms and technical details. It focuses on the concept and design of the business ecosystem, and the support and further development needs of the related technologies.

We hope that our investors, partners and communities will be able to come together and build this ecosystem together.

We recognize that VeChain technology may not be the most advanced in the world. VeChain had a good technical starting point and a cohesive technical team and continuous iterations based on the needs of the application. In the process, we will be very grateful if VeChain could contribute our discovery and breakthroughs to the community, industry and Blockchain technology.

2. Methodology and Technical Support

2.1 Methodology

Based on the understanding and following the objective of business rule, VeChain wants to begin with the smallest elements in business (people, object and money). VeChain wants to digitalize all of these small elements and build a general type of connection. VeChain builds the reflection on the coordinating activities of modern business through different smart contracts. It provides related value flow tool and system in order to create a new business model based on this

coordination pattern. After that VeChain builds a new kind of distributed business ecosystem that will be operated on Blockchain.

- 1) Digitalized the objective in a common way. The result by this digitalization can technically be accepted and used by any of the participant. VeChain uses the unified VID to mark the object and make a connection between the hashed data and VID in order to build the corresponded target data to VID. It also helps to IoT technology to complete the connection between VID and real life target.
- 2) To build a relationship type of connection with different object data by using the smart contract.
- 3) To use the abstract smart contract to cooperate relevant authorizations to composite modelling and customization, reflecting the different business activities in the business world.
- 4) A brand new digital asset (VeChain Token(VET)) that can provides the support of high speed value transaction.
- 5) Create a new trustworthy interconnect business model.
- 6) Different business model communicate and merge together to build a distributed business ecosystem.

Through this method, we can “translate” the target product, participants and business activities from the real business world into the world of VeChain. We can combine all enterprise, customers, and government resource and data information from different industries. In this way we can digitalize the cooperation and systematize the operation. So it can reduce the cost of the industries and even the whole society. It can also improve efficiency since resource can be optimized distributed and all kinds of brand new business model will be born.

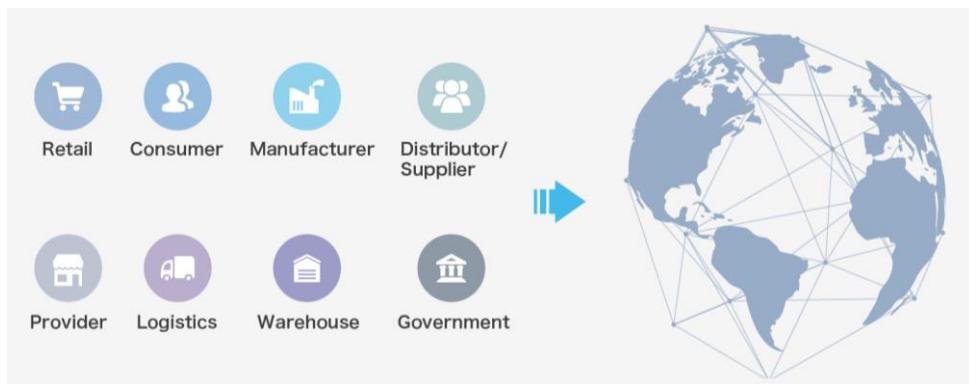


Figure 2.1 Standard digitalization in the traditional world

2.2 Technique Support

The path of VeChain's technology development is almost identical as the Blockchain technology. The initial idea was generated from the middle of 2015 and then VeChain started doing a series of technology verification (TPOC).

At the very beginning of the technology verification, we tried to use Bitcoin network (UTXO) to build the module and use Colorcoin to make VID come true. SideChain technology improves the speed of the trade performance so that the system can face the future business challenge. In the end of 2015, the smart contract of Ethereum has been improved step by step and we worked for it. We also made huge modifications and technical innovation to commercial requirement and Ethereum Fork based on the open source. Based on the customers and project survey, we kept improving the system in the September of 2016. For instance, we increased the data embedding and read performance to 300TX/s and we also improved the data security control in the bottom layer structure of the enterprise. In addition, we made joint development with CHAOS data management model, the IoT technology, the Blockchain technology and many different business smart contract.

Overall, the logic behind VeChain technology is always surrounded by business application. The idea of VeChain's management is practical requirement leads product design, product design leads product development and product development leads practical requirement.

In the whole process of evolving, our technical team got many supports from lots of great leaders, like the founder of the Ethereum- Vitalik, the founder of Jaxx- Anthony and many others. We feel grateful and appreciate of your open mind and the passion to the technology innovation

2.3 Technical Structure

Vechain's structure is based on application needs, and make standardized abstraction for every single technical structure layer. It enables every single layer to have the independent universality and let every model in each layer combine to each other efficiently. The standard unit model have tens of thousands combination of the application.

Below is the figure of Vechain's overall structure:

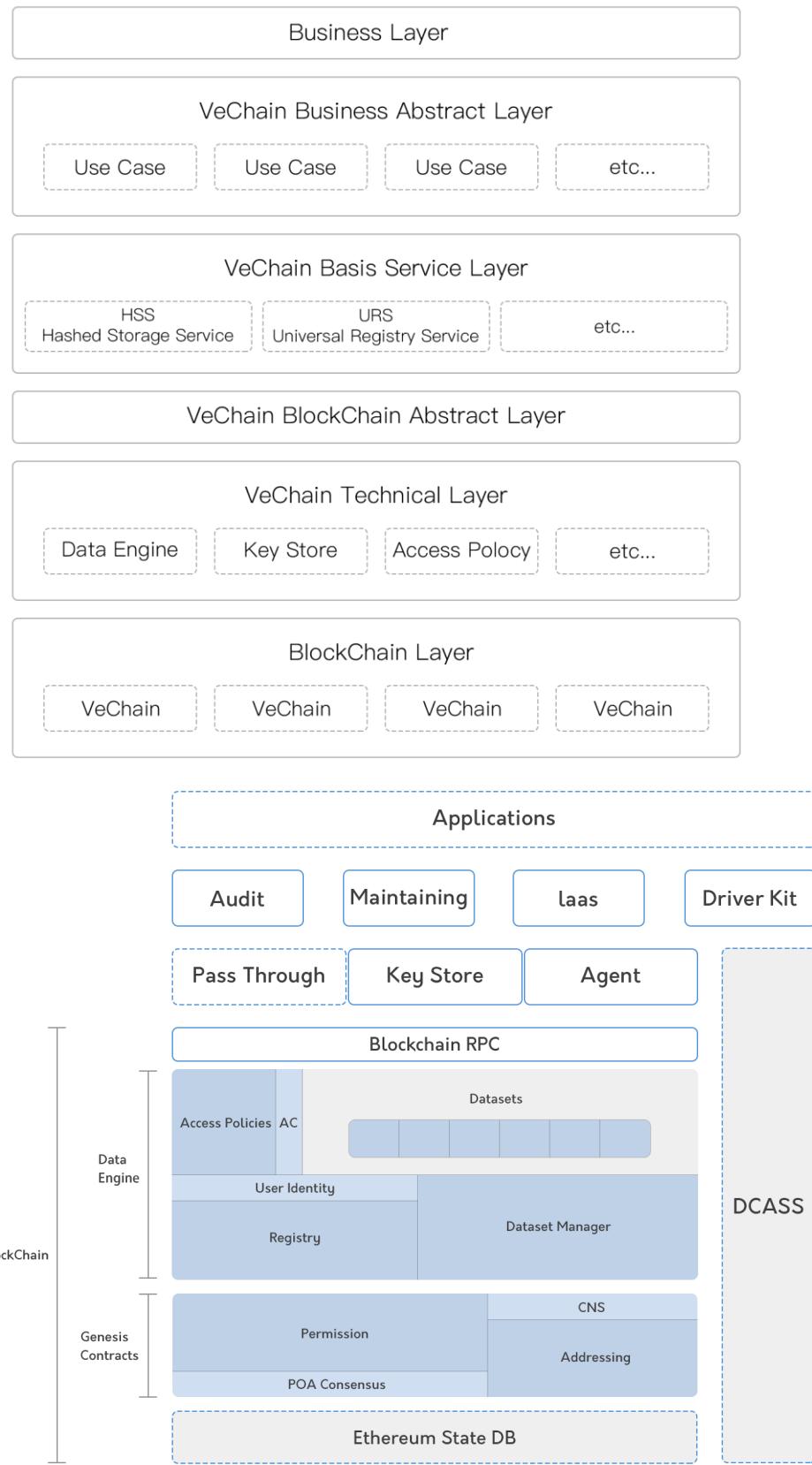


Figure 2.3 VeChain's technical structure

The structure is separated mainly by two parts of abstract layer: Blockchain abstract layer and business abstract layer.

Blockchain abstract layer:

- 1) It is the bottom structure of the basic layer. The technical focus is to fork Ethereum codebase and improve based on this, it includes:
 - a. DBGP- after maintained a certain level of security, then making a dynamic change for Block based on needs and requirements. This way it saves many storage space and system recourse cost. On the other hand, this protocol helps VeChain network to have 3 times better of efficiency than Ethereum after maintaining the security level.
 - b. DMBSP- combine with the traditional safety technique to cooperate with Blockchain's mining system in order to give enterprise level Blockchain application to provide data security protection.
 - c. DGIP- Implanting the data within the same category.
 - d. Under Development- -BLACP- storing Blockchain data with classification and use to differentiate saved data with different time and value.
 - e. Under Development - - PBCP- to distributed implanting and reading data with the same blockchain category.
 - f. Under Development - - DCCP- Syncing the data with different categories of Blockchain.
- 2) The upper level is the smart contract abstract layer for building a standard and modularity smart contract module (SSCU) in order to combine further, customized to face different industry, enterprise and smart contract for application scene (ASCM). Currently the smart contract inventory (VSCL) includes VID registration, data connection, status data implement, digital ownership, ownership transfer, authorization declaring, authorization transfer, multiple authorization and so on.
- 3) Building a Blockchain connector standard protocol (BGAP) to connect with upper business application layer based on the foundation.

Business application abstract layer:

- 1) In this bottom of this layer is the basic service abstract layer. The purpose is mainly to do a secondary operation for the smart contract of the bottom layer to build GBSM. It includes –hashing Storage service, a service model for CHAOS through URS. Meanwhile, this layer contains a special customized module for the data from the bottom layer. It includes index service for Blockchain browser, UDAS, HDMS, DCASS, CNS and DGS which includes the standard basic functions for smart contract on VeChain to save the time on deploying customized smart contract.

For this layer in the future, people will develop tools for visualized smart contract and through the service to build connections with smart contract. So even developers from different industries, or with no Blockchain experience

- can deploy and develop smart contract in order to push the application for the industry.
- 2) On top of this, the interface between the basic service layer and the business application layer is implemented for the two level application interface layer. The core of the development is standardization and to build the connection of business system that faces different types of data. In addition, accumulating more standard types by using application for many big enterprise that faces SAP, WMS and Salesforce, etc. As well as some common used website and mobile application connection.
 - 3) The top layer is business application abstract layer. It has standard application process module for different business scenes, different business practical developing module. So it can make the delivery and deployment for the development of the final application more convenient and fast. The developer of this layer does not even require to have any knowledge of Blockchain development so this can make more developers and technical service providers to use VeChain as the application of final customer development for Blockchain.

2.4 Achieve the Technical Details

VeChain's Blockchain is forked and improved based on Ethereum codebase. The basic technical index can see Ethereum's whitepaper as reference.

Below we are going to focus on discussing the technical application feature of VeChain.

2.4.1 VeChain ID Creation and Hashing

VeChain IDs are created by using a sha256 function which generates a random ID which is hashed before being written into a NFC, QR Code or RFID tag(s) to be used for each product.

All IDs are hashed by using a sha256 function which goes as follows: **SHA256(domain + '!' + ID)[12:]**. In which the domain is the qualified name of table that the ID settled. e.g. "**com.VeChain dbname tablename**"

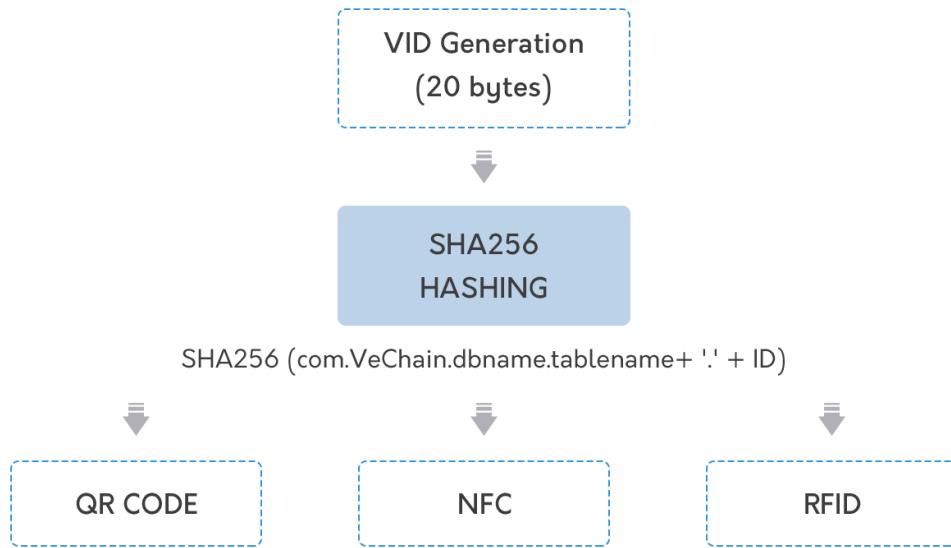


Figure 2.4.1 the creation of VID and hashing

2.4.2 Storage of VID on Blockchain

As previously mentioned, the hashed VeChain ID is written into tag(s) depending on the client's needs. After the tags are ready, they go to a testing process and are “activated”. Activation is done by using a custom-made software called “V-Operation” which can either run on Mobile or desktop operating systems. Upon activation, the ID is then written into Blockchain and replicated among all nodes.

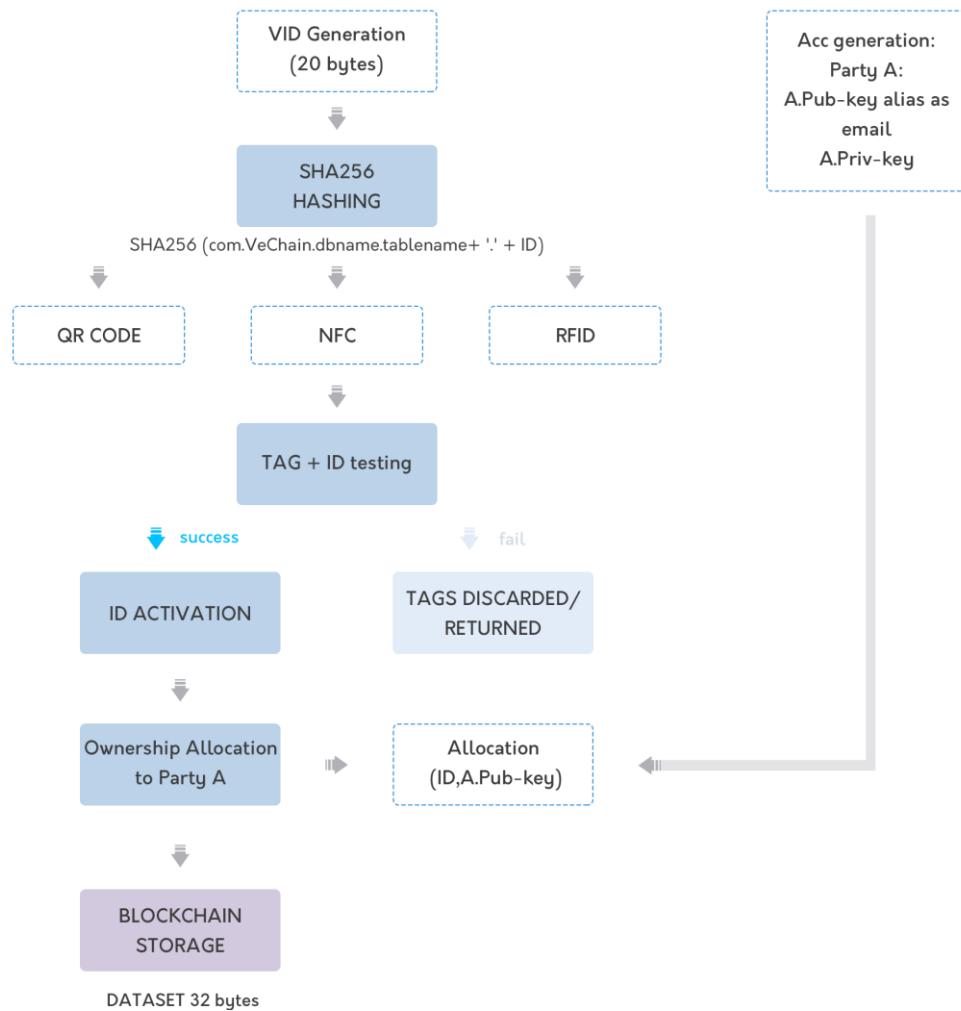


Figure 2.4.2 The storage of VID on Blockchain

2.4.3 Digital Ownership on Blockchain

VeChain uses a custom-tailored Smart Contract which enables **authorization-based digital ownership management**. The ownership of objects, represented by VeChain ID is linked to an account with the key pairs combined with public key and private key.

The public key is public and known as alias email address which can be recognized and accessed by anyone. The private key is to represent the authorization and access, just like a password, to the objects with the corresponding public key. The ownership management is to set a specific linkage between the objective ID and the public key of owner who controls the corresponding private key.

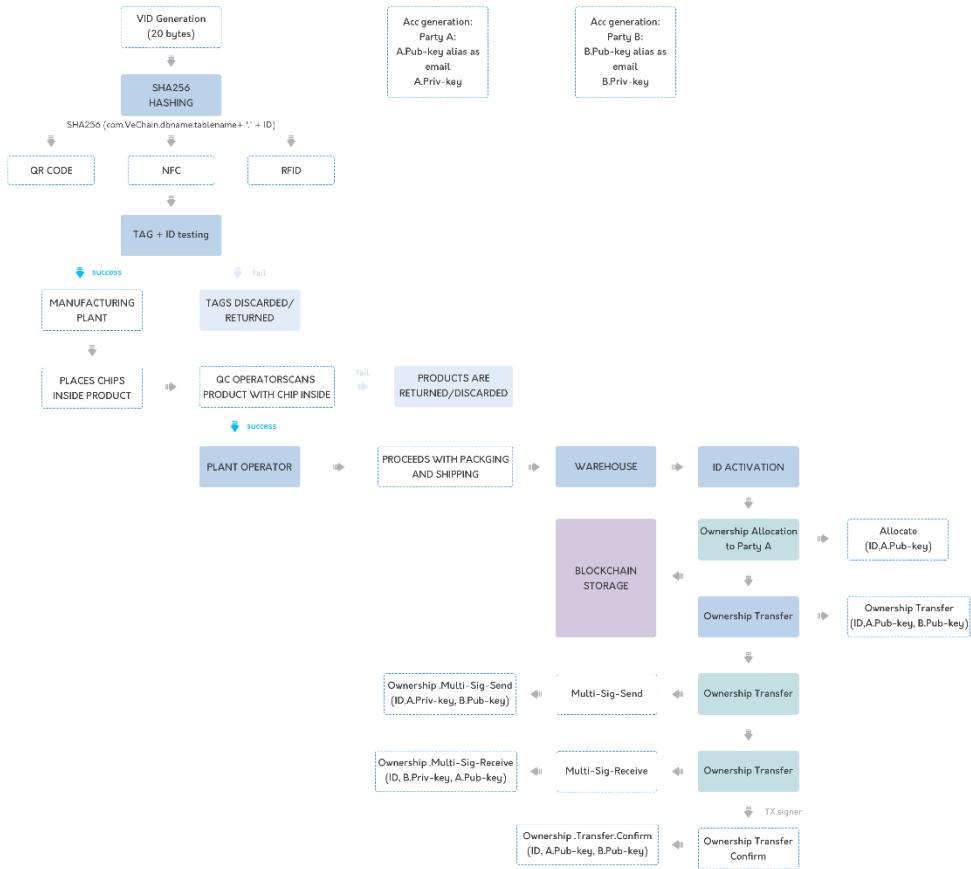


Figure 2.4.3 Digital ownership on Blockchain

2.4.4 Data Hashed Storage (proof of data)

VeChain accepts any type of data: (strings, numbers, booleans, etc). Data is identified by its hash (SHA256). Sample for accessing data via RESTFUL APIs:

- **Store data**
- POST <https://domain/hss/>

- **Retrieve data**
- GET <https://domain/hss/{hash}>

The data is self-verifiable. When the data is retrieved, it can be verified by comparing its hash to the hash provided.



Figure 2.4.4 Data hashed storage

2.4.5 API Gateway

Universal application architecture interface designed for complex processes. The API gateway is the main entry for all API requests, it encapsulates the internal structure of the application, and the client only needs to interact with the gateway without calling a particular service. When the internal structure of the upgrade or new features, the client is completely transparent, the client does not need to consider too many changes in access, only need to ensure that the exchange protocol is correct.

The following is about the network topology graph of the API gateway, deployment graph and functional graph.

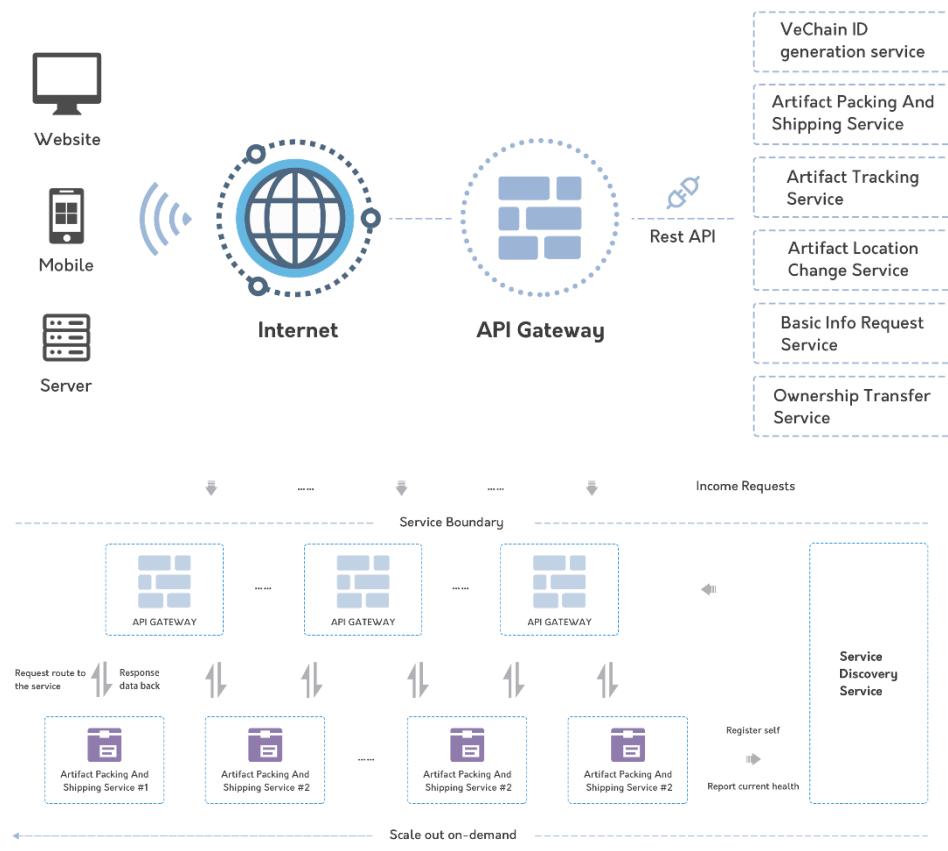


Figure 2.4.5-1 API Gateway-1

The resources of a server are limited, and the characteristics of the horizontal expansion make it possible for large-scale access. Different instances of the same service can guarantee a service request shunt by API Gateway. In API Gateway we can use different access policy like consistent-hash, ip-hash, random access or priority access. At the same time, API Gateway and Service Discovery Service also can be scale out on-demand.

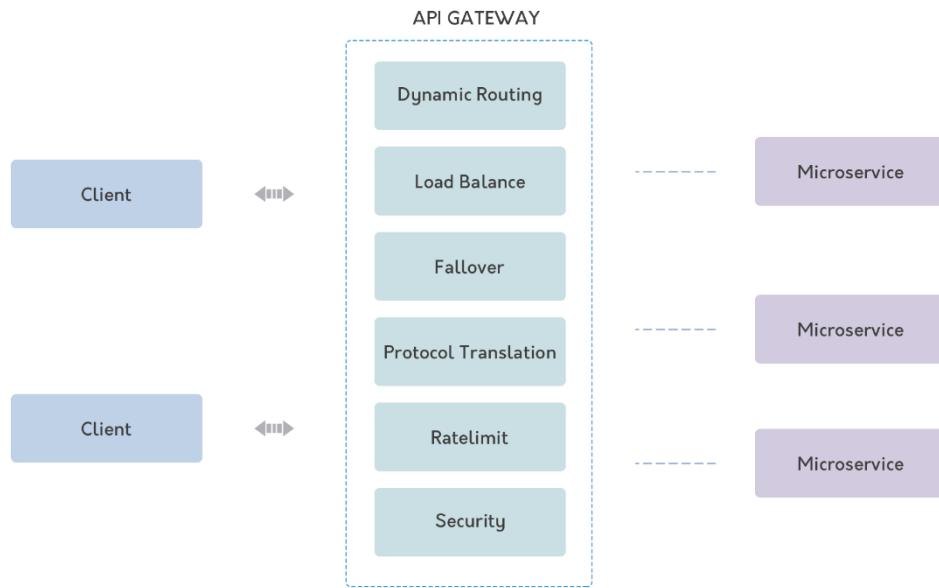


Figure 2.4.5-2 API Gateway

2.4.6 Service Discovery (SDP)

The API gateway needs to know the location (IP address and port) of each micro service that it communicates with. In traditional applications, it may be hard to connect this location, but now that it is based on cloud-based micro-service applications, which is not an easy problem to solve. Infrastructure services typically have a static location that can be specified by the OS environment variable. However, determining the location of an application service is not that simple. The application services are dynamically allocated, and a set of instances of a single service can also change dynamically with automatic scaling or upgrades.

Service discovery has two major modes: client discovery mode and server discovery mode. We are using the server-side discovery mode. The client makes a request to a service through the API gateway, the API Gateway queries the service registry, and forwards the request to an available service instance. The biggest advantage of the server-side discovery model is that the client does not need to focus on the details of the discovery, simply sending the request to the API gateway, which reduces the discovery logic that the programming language framework needs to complete.

The service registry is the core of the service discovery and is the database that contains the network address of the service instance. The service registry needs to be highly available and updated at any time. The service instance we use the self-registration mode. So the service instance needs to be responsible for registering and logging out in the service registry. In addition, a service instance

also sends a heartbeat to ensure that the registration information is not obsolete.

We choose etcd as a backend high availability, distributed, consistent key store for shared configuration and service discovery.

2.4.7 Micro-Service

Micro-service is the generic term for all backend VeChain services. This type of service can be customized according to the actual business interface to keep the separation between different businesses. Micro service can guarantee the service gray scale release, fast service upgrade or downgrade level. In our API Gateway ecosystem, the micro-service should provide below basic functions.

1) Register& UnRegister

Micro-service must be the initiative to register self to Service Discovery Service (SDS) when start up and must be unregister self when shutdown. SDS has 30sec to hold instance states, if unregistered when shutdown, after 30sec it will also be removed from service registry.

2) Report service health

SDS never know whether instances at backend are still available for serving. So microservice must report self-healthy in time and report interval must less than 30sec.

Micro-services are more complicated than traditional service, especially communicating between service and service at the backend. Currently service discovery service need instance to register self, so all instance need a logic that register to it. In the next time we should consider a 3rd party of register service for leverage. These service can deploy an instance of micro-service, and can config some information for it, can check instance health and report to SDS. So micro-service just can be consider a pure app for serving API.

2.4.8 Hashed Storage Service (HSS)

Hashed Storage Service (HSS) is a distributed storage service, which provides services such as digital files, pictures, text data and any other object-oriented reliable storage. Through the combination with VeChain, stored in various types of objects, HSS will ensure that the data cannot be tampered with. At the same time, the uploaded object, unless authorized by the uploader, otherwise it cannot be obtained and modified by improper means.

The HSS is mainly composed of two parts: object storage service and basic storage service. Object storage service is responsible for external interaction, save the object, access to objects and authorized object access; Basic storage

service is responsible for computing the object storage footprint, cut the object and the actual storage.

With the development of service scale, in order to system reliability, Data often need to be backed up. The number of backups per data is at least one copy, or even ensure the reliability of the backup, the number of backups may be two or more. This makes the storage utilization rate of only or less, each TB of data need to occupy at least 2TB of storage space. As data grows, the cost of replication becomes more and more obvious, with traditional replicas equivalent to at least 100 percent more storage overhead. Why distributed storage can help you setup a highly-available storage system with a single object storage deployment. With distributed service, it can simplify the data backup program, reduce disk space, but also make services available, improve data's availability and durability.

Erasure Code & Reed Solomon In the storage system. The erasure code technology is mainly through the use of mathematical algorithm to verify the original data to be verified to achieve the purpose of fault tolerance. It can be used to reconstruct missing or corrupted data. Reed-Solomon (RS) code is a storage system that more commonly used in an erasure code. Our Storage Service use RS algorithm to shard objects into and parity blocks. You can lose as many as drives (be it parity or data) and still reconstruct the data reliably from the remaining drives. Amazon S3 Compatible the APIs In the low level is compatible with the Amazon S3 API.

Compatibility will lead to data access bonuses. Most developers are familiar with Amazon's S3 service and are familiar with how to use its API. A compatible interface reduces the possibility of external access.

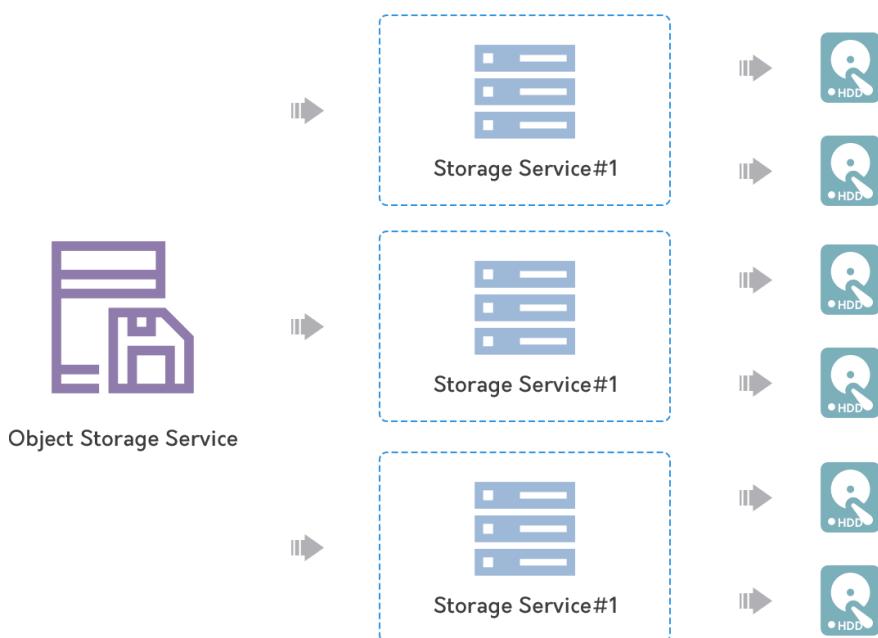


Figure 2.4.8 Hashed storage service

2.5 Blockchain and IoT

In 1999, the concept of Internet of Things (IoT) is proposed by a British Scholar Kevin Ashton in MIT. After discussed with many senior level managers from different enterprise, Ashton defines IoT to:

"A network containing all 'smart' devices with some sort of sensing mechanism that can communicate via the internet with other smart devices or the cloud, without human interaction."

We can see IoT with mainly two parts – Perception and Connection based on this definition.

- ✓ Perception is the idea of digitalized the real world data, like temperature, humidity, similar environment type data and personal ID.
- ✓ Connection is sharing the digital information through Bluetooth, WIFI and mobile internet.

In the future, the trends of IoT will be widely used, more different types, high sales and fast development. It will have a huge impact on the global industries. By quantity, the sales amount of IoT equipment will increase as a rate of 15% to 20% in the near future. IDC predicted that sales amount of IoT equipment will hit 45 billion in year 2020.

2.5.1 The Issue of IoT

The issue of IoT technology has been discovered in the nineties. Until June of 2016, 3GPP announced "release 13" and defined the IoT connection standard. This solves the four issues of IoT: Limitation of the connection capacity, limitation of coverage area, low standby time and high cost. Since September of 2016, every mobile device manufacturers released IoT connection plan for business. This focus the concept of different application can choose eMTC, NB-IoT, EC-GSM and other different technology.

With the new IoT standard, it has the feature of great connection, huge coverage area, low energy waste and low budget cost. Since then, IoT industry started to grow rapidly. However, we believe there are still three major issues for IoT: fragmentation of the standard Communication protocol, high cost of development and maintenance and lack of privacy, but these issues can be solved by Blockchain perfectly.

2.5.2 Blockchain and IoT

Currently, there are many exploration on different IoT and Blockchain application on smart system. When those are applied on IoT, the concept of IoT opened up a road of innovation with unlimited possibilities. The Blockchain technology can be used as tracking the usage history of different equipment. It can also help to complete the trade with different equipment. This technique can provide the data transaction within different equipment that makes IoT equipment independent.

Blockchain will realize the self-management and maintenance of the equipment. It saves the huge maintenance cost of cloud system and reduce the maintenance budget for IoT equipment. The private key that made by the equipment will ensure personal data won't be stolen by strangers. It increases the safety level of IoT and the economic effectiveness with the combination of these two concepts.

We believe that the IoT technology and the Blockchain technology cannot be split in the application. From the view of self-development for IoT, IoT wants to build a world with everything connecting to each other and this process requires three steps:

- 1) It requires a unified communication standard between one thing and another. It means the equipment can communicate with the same language no matter where this equipment was built. IoT standard that published by 3GPP provides a physical channel for thing. In addition, the Blockchain technology provides logic language for thing and this makes different types of IoT equipment communicate with one unified language.
- 2) With the foundation of the unified language, personal Identification will be required during the communication between different things. Then it needs a standard identification code system to support the unified language. This means the Blockchain technology is the best solution to unify different manufacturers and the identification code will not be controlled by anyone.
- 3) With unified language and personal identification, the connection between those two will require more cooperation and business activities. This means smart contract needs to be built and digital currency will become the transmission carrier since the value needs to be transferred during the cooperation at the same time.

As a technology that provides the service of trust, Blockchain can ensure the true effectiveness of data on the Blockchain network, IOT is the key to ensure the true effectiveness of data when it's been uploaded from the first time from the original source.

On the one hand, IoT helps to establish the congruent relationship between the real physical world and Blockchain world. On the other hand, the IoT technology can reduce the disturbing factors from the source to ensure the true effectiveness of data.

2.5.3 VeChain and IoT

VeChain technology team contains a very important component – the IoT technology team. They focus on the responsibility to coordinate the IoT development with Blockchain application, which includes:

- 1) Encrypted chips tag technology development.
- 2) The identification of IoT sensor and data security.
- 3) Security and authorization module of NB-IoT.

IoT equipment is very complexed and we need to classify them with different point of views:

- ✓ From the point of view of power supply
 - Equipment with power source: equipment with battery equipped, like temperature sensor, GPS.
 - Equipment with no power source: equipment with no battery equipped, like NFC.
 - Mixed equipment: Equipment with battery equipped and can also get power from other places.
- ✓ From the point view of communication distance:
 - Close range distance equipment: resolve the communication within 10 meters distance, like NFC(1 meter), RFID(10 meters), bluetooth(10 meters), etc.
 - Middle range distance equipment: resolve the communication within 1 kilometre, like WiFi, sub 1g, lora, etc.
 - Long range distance equipment: resolve the communication with more than 1 kilometre, like NB-IoT, etc.

We upgrade traditional IoT equipment on the chip layer and put personal identification with asymmetric key algorithm.

- ✓ **Personal identification:** Every IoT equipment requires a unique identification on the network and this ID can be identified by other participants within the network. We enciphered each different equipment or object to ensure the code is unique and cannot be recognized.
- ✓ **Asymmetric key algorithm:** asymmetric key algorithm is the foundation of the internet and the important feature of Blockchain. It can identify and authorize the equipment based on the identification of equipment. Through public and private key algorithm, we can identify if the equipment could connect to the network, if the digital data source were reliable, or if smart contract can be operated. To ensure the safety level of asymmetric algorithm, we will put the private key in the safety area so it cannot be read. In addition, asymmetric algorithm runs with the processor's safety mode and it can ensure the safety during the process of calculation.

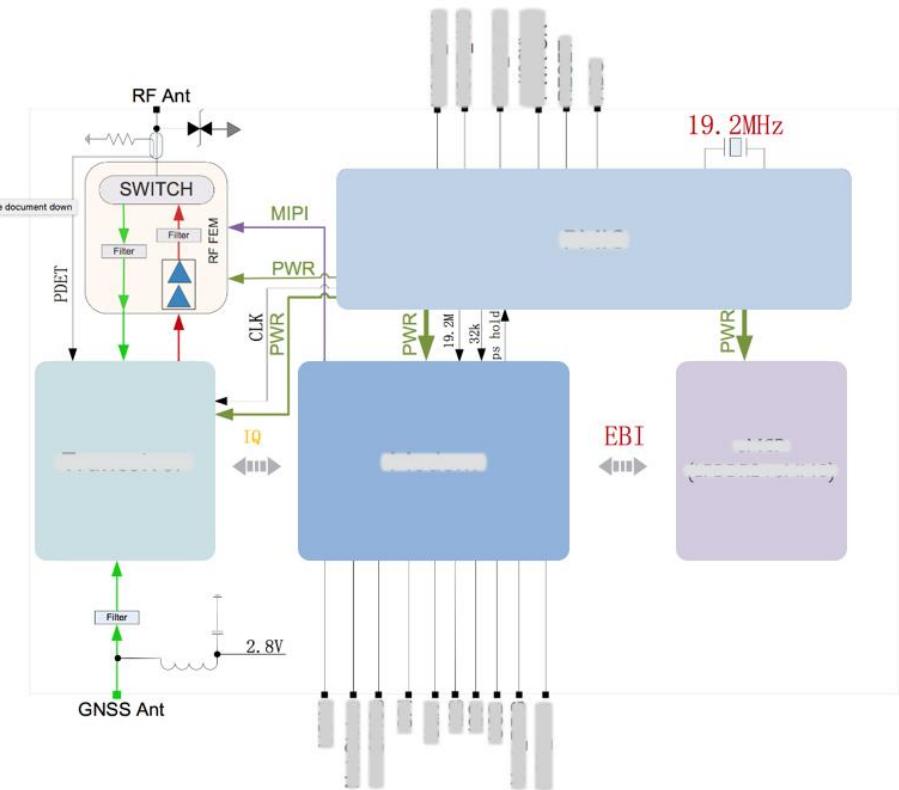


Figure 2.5.3 NB-IoT safety module

2.6 Technical Testing

VeChain team follows the procedure of professional software testing. Software should be predictable and stable and meet the standard of the product so there should be no unexpected results. With the digital information keep expanding, complicating and smart, the development of software testing is improving as well. With the maturation and systematizing of theory and practical, it shows more impact during the software quality check. General experience suggests that in a typical coding project, software and system testing cost about 50% of project time and over 50% of total cost.

The most concerned question that software testing care about is in which subset we can find the most problem during all the possible testing. We can put the testing into three category:

- **White box testing:** also named structure testing. This method regards testing software as white box. Based on the internal structure and logic, software is designed as testing sample to proceed a testing for the program's path and process. White box's main technique includes statement coverage, branch coverage, design coverage and elementary path coverage.
- **Black box testing:** also named functional testing. This method sees testing software as black box. Without considering of internal structure and

characteristic to test software's external's characteristic, black box technique mainly includes equivalence class partition method, boundary value analysis, cause and effect diagram method, status mapping method, measurement method of outline and many typical malfunction model.

- **Gary box testing:** This is the test between white and black box testing. Gray box testing mostly used on integrated test phase. It focus not only the validity of output and input, but also program's internal condition. In addition, Gray box testing is not detailed and completed like white box testing, but focus more on internal logic than black box testing. It usually estimates the operational status through certificate of phenomenon, event and symbol.

Team VeChain sets a special testing team and take on the role of "quality management". The purpose is to make corrections in time and ensure the smoothly operation. Thus software testing is mainly for verification and confirmation. The target for software testing is not just program testing, but also includes all the documents from different phases of development, such as testing guidance book, testing project plan and testing report.

- Testing guidance book: describing the testing requirements and theories
- Testing project plan: describing testing sample and testing methods
- Testing report: output testing results

Testing is a process of convergence step by step and it strictly follows PDCA quality circle. The testing process from the description above is only one part of the PDCA. PDCA is acronym for Plan, Do, Check and Adjust. The PDCA circulation perform the quality check and it will keep going like this by using this order.

- P (Plan) includes the confirmation of the testing plan that contains unit testing, integration testing, system testing (function, performance, safety and compatibility) and examine testing.
- D (Do), based on the testing plan to perform the test.
- C (Check), the final testing result and reflect the feedback to development team.
- A (Adjust), development team improve and fix the original code based on the testing results.

The goal of VeChain's software testing:

- Lower computer(PLC): Embedded software of IoT parts
- Client: PC end, mobile end (ios, Android)and terminal software
- Cloud
- Server: software on Website and server
 - Blockchain part
 - Smart contract part
 - Interface part

For individual and interface between different parts on the four goals, we have professional testing team defining the testing guidance book, testing plan, and output result report. In addition we will use quality management methods by PDCA to complete the software testing and ensure the quality of VeChain's product.

This is part of the pressure test data result:

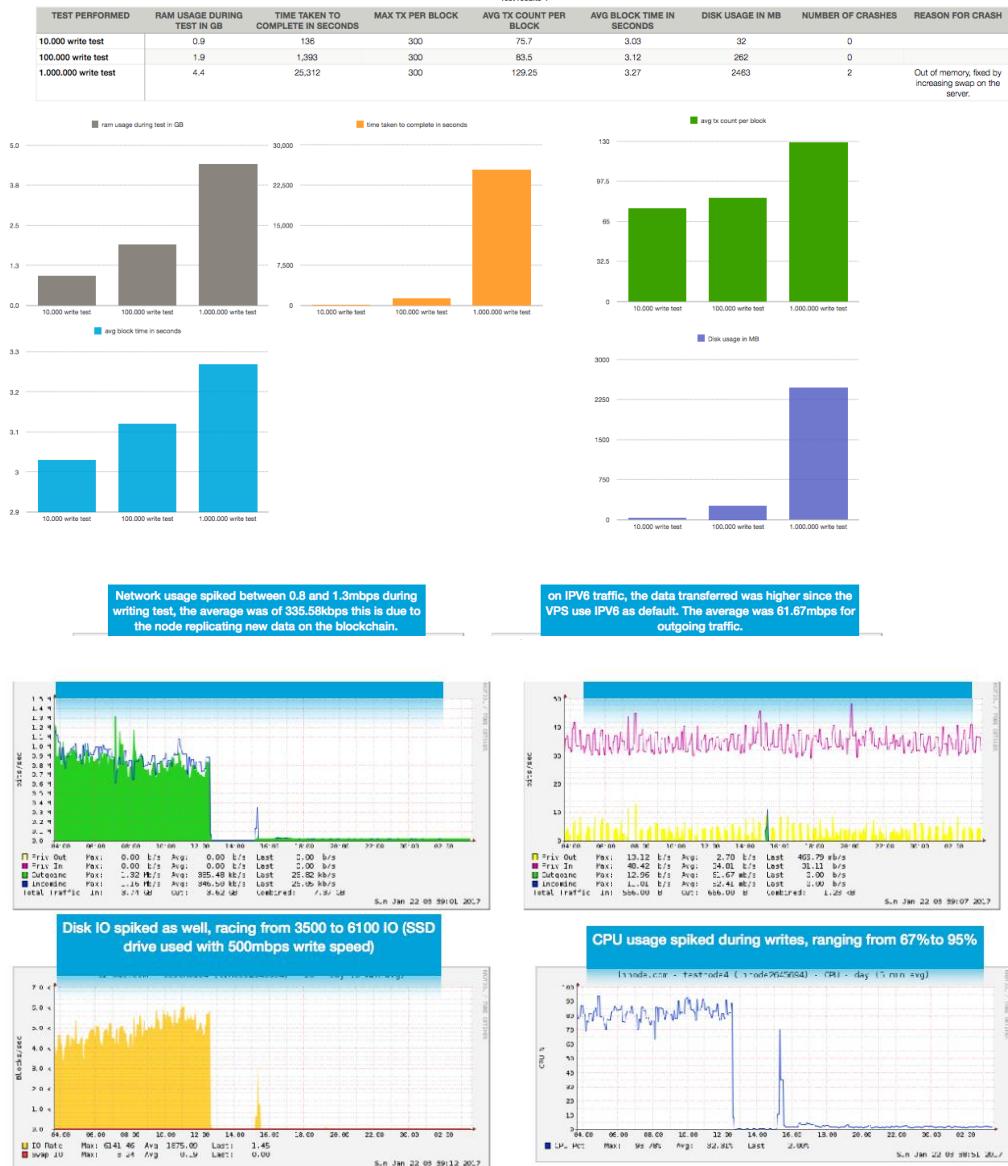


Figure 2.6: Part of the pressure test data result. Testing environment is using the lowest cloud server configuration with different locations from global. The basic set up is single core 2G server.

2.7 Technology Development's Path and Plan

The development of VeChain technology has been through two years and the core of development focuses on three areas: application, standardization and

safety. VeChain team will keep following these three basic ideas to continue the development.

VeChain technology team has three units:

- 1) R&D – focus on the bottom level of the technology and development, as well as with the newest technology analysis and experiment. In addition, they will make plans for the next generation's possible path and feasibility analysis.
- 2) Development – Based on the result of R&D, to perform and complete the development and get the initial testing result.
- 3) Testing, deployment and maintenance – Based on the development result, R&D need to improve and correct the test result as well as taking care of related deployment and maintenance.

Below is the path of the VeChain technology development and future plans:



Figure 2.7-1 Technology development path

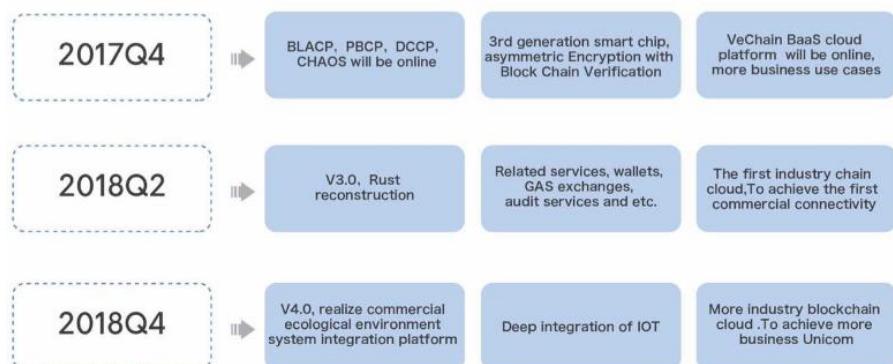


Figure 2.7-2 Future technology development path

3. The Industrial Application and Expansion

In the past two years, VeChain has gained great amount of experience from many different fields of the work and some clients are world famous firms. The figure below shows the application structure of the VeChain:

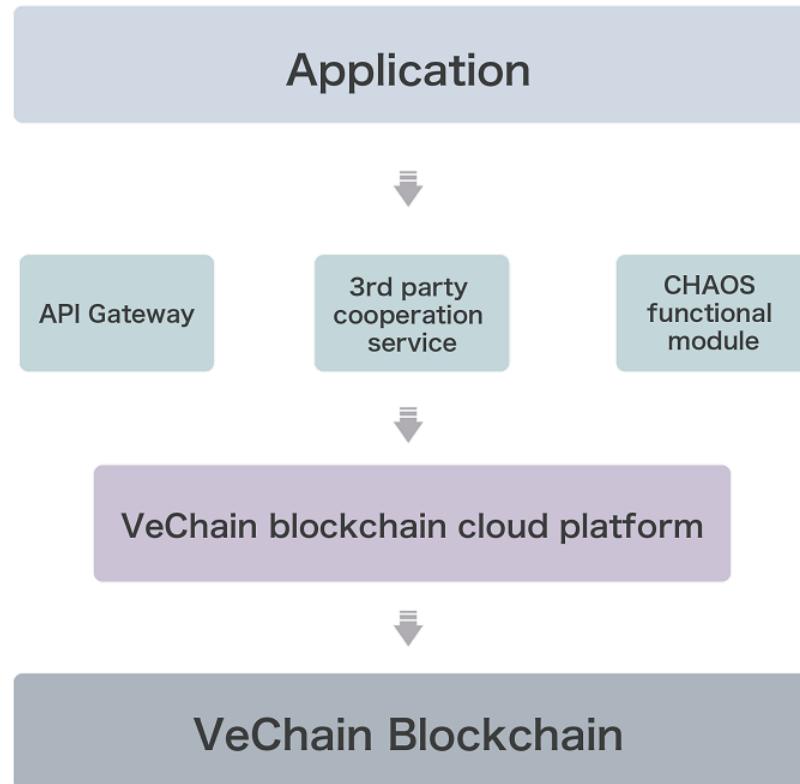


Figure3-1 Industrial application structure

Based on the bottom structure of Blockchain system, VeChain loaded a platform that is convenient to use for the business partner and a one-click deployment Blockchain Platform. Users only need to select the industry they are in and the solution plan they want, the system will allow the user to manage their own Blockchain nodes, smart contract and generate related API setting.

Meanwhile, together, VeChain and cooperative partner are developing a third party application which is focused on Blockchain to serve VeChain's customers. For instance, VeChain is developing a unique Blockchain audit service with a world famous accounting firm. It provides data auto-collection and audit suggestion on Blockchain overall operating states, smart contract status, business operation status.

VID	UID	Date	SC ID	Hash
6607de2416787b7c92aefbc02910cb2789e2	16520a127a187fb1c66a1e220bae015	3/24/2017	baf57ec2789674ec8c8d94bc0cf0f4e5d500859	14bf14a42497e8d13303a3a7febe411e23ab9c83e729c58619a2333d399
78849977e2730d2b7e1314987a6dc5a99c9230e	bfc1cd637501b181364102dca2b5f	3/5/2017	5a2030edeb350fa0d2483d9a12999d496d7272	b294b95813fb44741e3a4e51506d14b4d441aa3e40465a2001e1ca
1d4b1db09445db722bd2a0b4b747e3c631	04bd05927d3bde6c79f181d4b60033	4/2/2017	8649-086f01a951934726548a35ee4b8a3d4	4-d9xa358h4e4688866a8e1894720554b6e8881d18488a7a6903d934fb
9b206a8d1e402b03ea8422ac1ef49259354	2c537e005116323e5e556a025d0f	3/19/2017	c9426f0d1f16cc6a1a2b3e3c31c359c59b40	70a6397fbcc0c5e15a1999784a41e1a0588643774a0411
b594a303913754744744aae5b4a0a9471b	abca5d8ce32620f0161e2f2232948	5/17/2017	f1875eew097983e6e8511730484914463c	198f8a771e23b723991f42d13a5f9460a034993e96783a49914892
5775a27174d645b7c7446e015fe4a3e29716	23d3540a48739ew7b3d65949733662e	3/8/2017	029a536a07115494c41fb15d4e42230115	a217beff021391904725a9e4a0c509f200a3d117932
c95c78779257ea6d62b7913e10a264058275	5594b706ew267301912c37ab38626	3/24/2017	c0ea997770e753443e9773beecf1481177925e5	395ef41940a5ed3e0cabc1f057e14393421171a332c5d375d8464a40
6b344259b50717a6927fb7e866623a933a466ca	353b6edc2180e48323d4f92e8173	3/16/2017	811e9799e8d8ab4b6b506233a2a9b9e6117	c09fa10f98212137a1446cc9ec05194241973e2212a29e0d922b6b
0e7d2190227508e6922171713465030fb8a729	e222979703433c0f8e36a09467e0d5	4/15/2017	2723e70d4099ua028bae4315427e0233928	13/13207a3403a22236e8f27752a6a61408e77397e754f19516a19196754
b036fb240419585050e3b39149648c754927b	6098f5334a50503716b7e1fa511ee	3/7/2017	64aa4c65d6e06e95x377e5431e4373971996a01	1a3b0ew18d0d239956a4e4c74911363w5c624617528452695

Figure 3-2 Third party service application

In addition, the application structure of VeChain also includes self-developed distributed encrypted database service- CHAOS, user private key management and smart contract authorization, etc. This modular service model makes customers and service provider's development more convenient and flexible.

VeChain plans to use these successful cases as template to expand and develop more quickly. VeChain wants to let more enterprise and business activities operate based on the VeChain platform. In addition, establishing the connection with these business activities step by step. At the same time, through developing related business smart contracts to promote the circulation of VeChain Token(VET) in order to complete and expand VeChain's distributed business ecosystem step by step.

3.1 Fashion and Luxury Industry

According to the survey from year 2015, it costs fashion and luxury brand of Europe 9.7% of the total sales every year, about 2.87 billion dollars, for anti-counterfeiting. Due to the overrun by the fake goods, it costs Europe lost 363,000 jobs in fashion, manufacturing and retail industry each year.

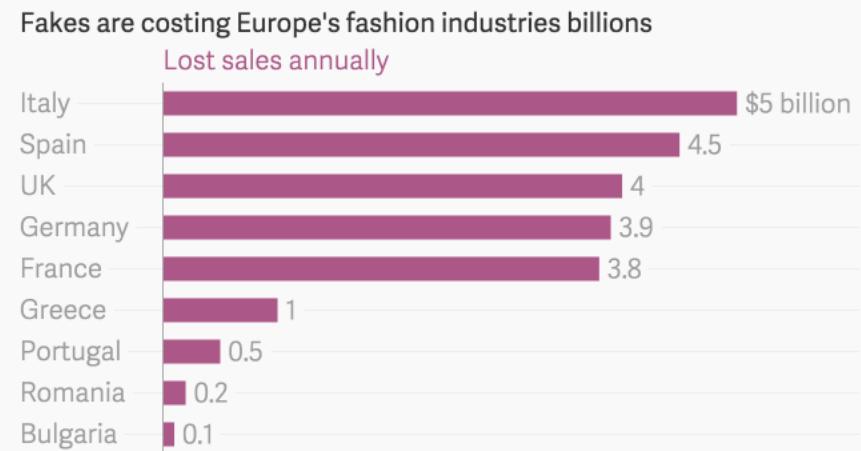


Figure 3.1-1 the impact of fake good to the Europe fashion industries

VeChain aims to focus on this industries by following production management, production channels, anti-counterfeit and the connection with customers. The third party makes product based on the order plan and at the same time, the brand party can “activate” this product when verifying it. This is the process of Chain to verify the enterprise’s SAP order form and authorize the real thing from the beginning of the process.

Meanwhile, VeChain establishes the data connection of WMS with the enterprise and complete the Blockchain process with dealers and retailers sales channel, in order to achieve the management to the sales channel. For the customers, VeChain provides a digital ownership to build a bridge between the brand and customers and keeping the transmitting to CRM and after-sales service. In order to establish a customized individual service, after-sales service and customer care for customers. It can even track the trends of the second-hand market. At the same time, the anonymous privacy protection feature of Blockchain perfectly fits the safety rule of EU’s GDPR (General Data Protection Regulation).

VeChain makes business with famous Europe luxury brand by putting IoT encrypted chips that is based on the Blockchain technology into product. As the medium of the data collecting for Blockchain, the chip records the every “logistics”, warehousing and transmission. After the customers get the product, they can scan the chip behind the tag through the APP and they can know the “history” of this product. So the customers can identify the authenticity of the products, and making a statement of the product’s digital ownership at the mobile end.

3.2 Food Safety

Food safety directly affects people’s health. Since the budget of being healthy has growing all the time, both producers and customers would care a lot about food safety. However, traditional food safety relies too much on the process control and

entrepreneur's sense of responsibility. So it is pretty hard to ensure the safety of food by using automated methods. It is quite difficult to track the original source if any problem occurred.

Yet the Blockchain technology could bring safe and reliable solutions to the food industry. The Chinese government has already confirmed that food certification and tracking through supply chain are the key steps to find and eliminate the source of pollution in a very fast way.

The Liquor tracking platform in Waigaoqiao free trade zone, which built by VeChain, can track the liquor product from the very beginning of the process, even when the liquor was still in the overseas winery. This is the first successful case in domestic. Every details about the bottle of red wine is marked and recorded at the beginning of the process. This way company can use smart contract to track the whole life period of the red wine, from the warehouse in the free trade zone to for the first time. From the distribution center and finally reach to each different sales channels and stores.

Customers can identify and track the information of the wine through the in-store touch-screen or even customer's own smart phone. High-end wine are also equipped with IoT Chips with the feature of safety and convenient. Customers can use their Cell phone to check these information as well which increases the security level.





Figure 3.2-1 VeChain application in the wine Free trade zone: background management system, smart front terminal, mobile showcase

For the next part of the plan, we are going to let more oversea wineries and welcome more imported product providers to join into our platform, so customers can feel safe about the product. We are also going to open the connection and start the cooperation, let the customer see more information about the product.

On the other hand, the product that VeChain is following right now is one of the most focal point product: dairy, dairy food safety is a hot topic that people pay huge attention to. The regulation requirement is stricter, especially when the news about domestic milk powder contains melamine and makes the milk powder toxic. Mengniu Cooper has to increase the breadth of the product sale and the multiple security check to ensure the source of the dairy product is safe. This is what they paid for the huge loss of customer trust. In 2007, Mengniu corporate spent 3.302 billion Yuan in sales and distribution cost, and the number increased to 4.428 billion Yuan in 2008. The advertise expense in sales percentage has increased by 2.1 percent to 9.3%. This is merely the cost for losing trust of one single enterprise. Although this event has passed for so long, the dairy industry is still paying for this. We can definitely make more example of this, it shows that the cost of trust is almost unbelievable.

VeChain provides the standard adding Chain function that let all participants upload the data by using the permission to complete a Chain. (Like Figure Below)

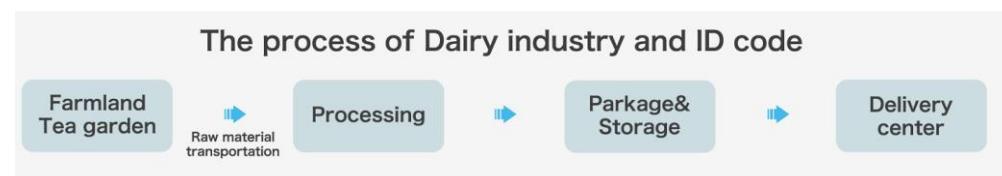


Figure 3.2-2 The process of Dairy industry and ID code

VeChain can help Dairy Company by providing farm information, including Fertilizers management, the audit for feed supplier, the healthy condition of the livestock and the drug use on the cow and environment report. This process ensures the food safety from the very beginning the milk source. VeChain can also help the production supplier to check the receiving time of the raw milk, the storage condition of the raw milk, production reference number, and the detail information of processing personnel and people who are responsible. In addition, Package storage can use the technology methods to tracking the temperature and humidity while transporting. Moreover, the production's loading information of distribution center and data information can also supported by VeChain. Finally, owning these information cannot just prevent counterfeit product but it also increases the product supplier and customer's faith for the industry.

3.3 Car Industry

Industry Chain for the Car industry is very complicated. There are many participants, like manufacturer, different agents, regulator, financial service provider (Insurance, Bank) and personal account. In the life circle of a car, there are a big portion of the “user data” are never owned by customer, instead these data are separated in the pocket of different participants. This causes many difficulties of car information collection and verification.

VeChain and Europe strategic partner Visco, Microsoft invent a verification idea for many international car enterprise together. In the project, VeChain team is responsible for completing the Blockchain deployment on Azure, developing and deploying smart contract and provide standard API to the upper level developers to complete the final product.

Digital maintenance principle: Every car can build their own digital record and build the authorization of the ownership. After car owner bought the car, they can use authorization and non-authorization feature to give permission by the server maintenance suppliers. So every single maintenance data is recorded to the Blockchain. This way the data provided by different maintenance service provider can build the real grouping record step by step. For instance, insurance, bank and other financial service provider can provide fast insurance and value assessment based on the data supplied by this trustworthy network. The operating expense that saved by this can be returned to the car owner.

“Green Driving”: Green driving is the shared electromobile project provided by this car enterprise. Every customers can record related driving records by internal computer in car and upload them to the Blockchain and connect the data together which owned. In the future, these data can proves the beneficial information of the certain project like “carbon emission” and even customer data can be used as

individual creditability.



Figure 3.3 Showcase of VeChain usage by a famous car enterprise

3.4 Supply Chain Industry

Traditional supply Chain includes: original material supplier, manufacturer, agent, logistics, customs inspection agency, storage, retail and finally customers.



Figure 3.4-1 Traditional supply chain

Traditional supply chain industry is facing many problems includes:

- 1) It is quite difficult to track supply chain since its cross-region
- 2) It lacks of transparency between different supply chain and information
- 3) Data security vulnerability in different enterprise of supply chain
- 4) Money flow transport has bad timeliness

VeChain provides Baas (Blockchain-as-a-service) service to one of the biggest freight forwarders K+N to track and manage all the products from many world's famous brand. To ensure the data protection and privacy as the precondition, VeChain completed the connection with different customers through a common service platform. The operation staff can complete related business work by directly using the handheld terminals.

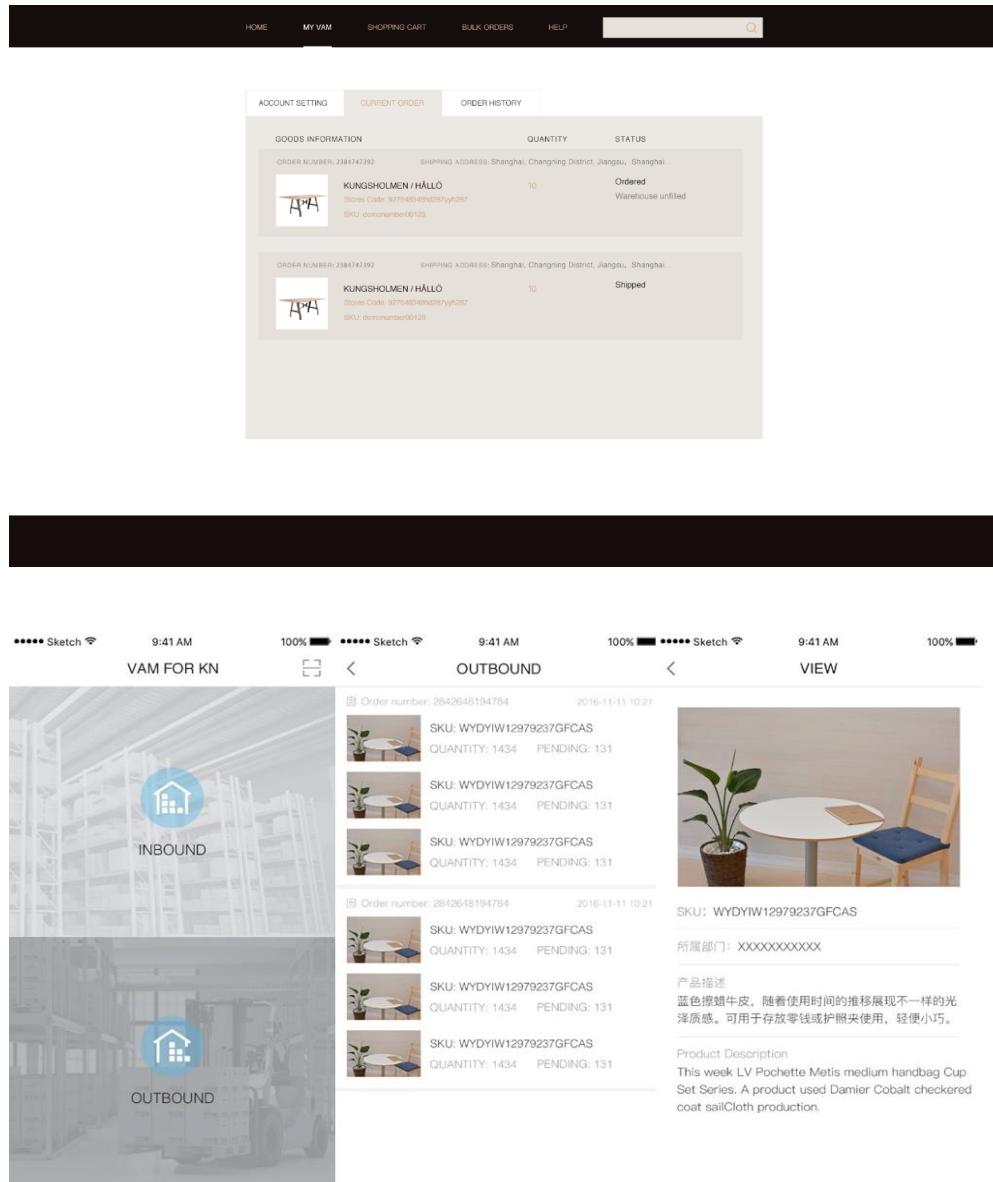


Figure3.4-2 The showcase of logistics issue solution by the famous freight forwarders

In the late period of the plan we are going to make connection with more related cooperative partners, service provider and regulators.

3.5 The Agricultural Industry

The Chinese market is facing many critical issues like the scale of the agriculture

which is too small and separated, the quality of the product is uneven, the lack of the safety level of the product, low productivity and environment pollution. It is quite hard to fix the issue completely by simply using a certain technology from the internet or a law regulation provided by the government. We can only change the thinking model by using the technology like the special Blockchain cloud project that is exclusive for the verification of the green organic agricultural in order to use industrial management to build modern agricultural.

China is promoting Agricultural Cultivation Management Plan by using IoT technology, agricultural planting process management, the Blockchain technology, big data and AI (artificial smart) to complete the management of the process before, in the middle and after the agricultural production. In this way, good currency drives out the bad currency to achieve standard agricultural market.

With this background, VeChain is cooperating with PwC, China Unicom and Liaoning academy of agricultural sciences to develop the special Blockchain cloud project that is exclusive for the verification of the green organic agricultural.

In this project, VeChain has registered the greenhouse for every farm by using the Blockchain tech to build a data model to record the functional data of every greenhouse. Data source has two main parts: the first part is the production operation data which recorded by the famers directly; and the second part of the data comes from the IoT sensor in the greenhouse. Based on the combination of the data and risk assurance service from PwC, it will establish the foundation of the trustworthy data for the green agricultural verification by the academy of agricultural sciences. In addition, with the support by the IoT equipment, it improves the efficiency of the farm work by about 9 times.

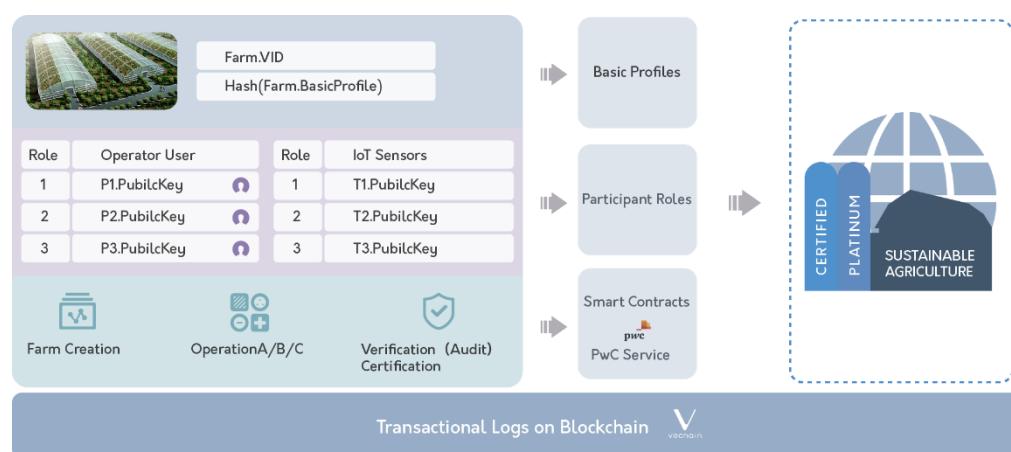


Figure 3.5 Application of agriculture combines with IoT technology

3.6 Blockchain Government Affairs

Government agency shows interest in the Blockchain technology. China ministry

of industry and information technology has released a white paper about the application and development of the Blockchain technology. The State Council underlined that Blockchain can bring a world with trust.

The science office of British government has reported the potential qualities and advantage of Blockchain technology in the recent report: "Distributed ledger technology has the potential to transform the delivery of public and private services. It has the potential to redefine the relationship between government and the citizen in terms of data sharing, transparency and trust and make a leading contribution to the government's digital transformation plan."

VeChain has signed strategic cooperation agreement with local government with big data collection to build case project for Blockchain Government affairs. VeChain has a targeted plan in some very typical application area of the Blockchain technology.

For instance, commodity inspection is always reported by manpower to the inspection agency, then agency will inform the client for the material of random sample. Because of the system that agency use is so different than the others, also agency and client are only using manpower to report and delivery every information transaction. This may cause the unmatched information, very long process, low efficiency, and even the risk of data manipulation.

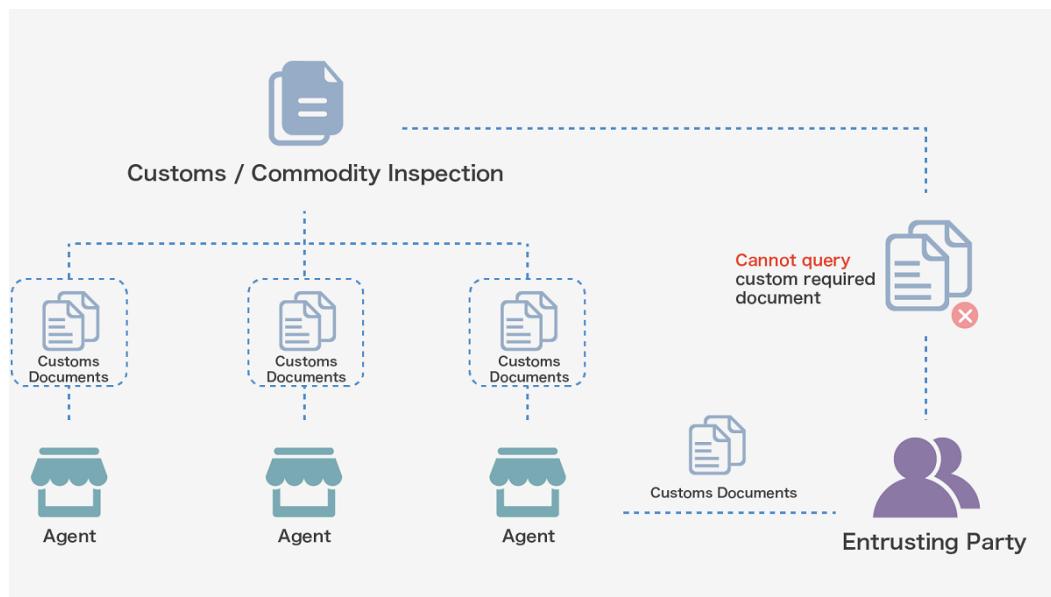


Figure 3.6-1 Blockchain Government application

VeChain will share related data on the Blockchain platform, agent and the client both can check the random sample date through VeChain's APP. This makes the whole process paperless and complete the connection between Blockchain ID and waybill number, data ID and waybill number, then finally smart contract ID

and the related operation function. The goal is to reduce the paper use from 20-30 pages to 2-3 pages per file. This will increase the efficiency by 80%.

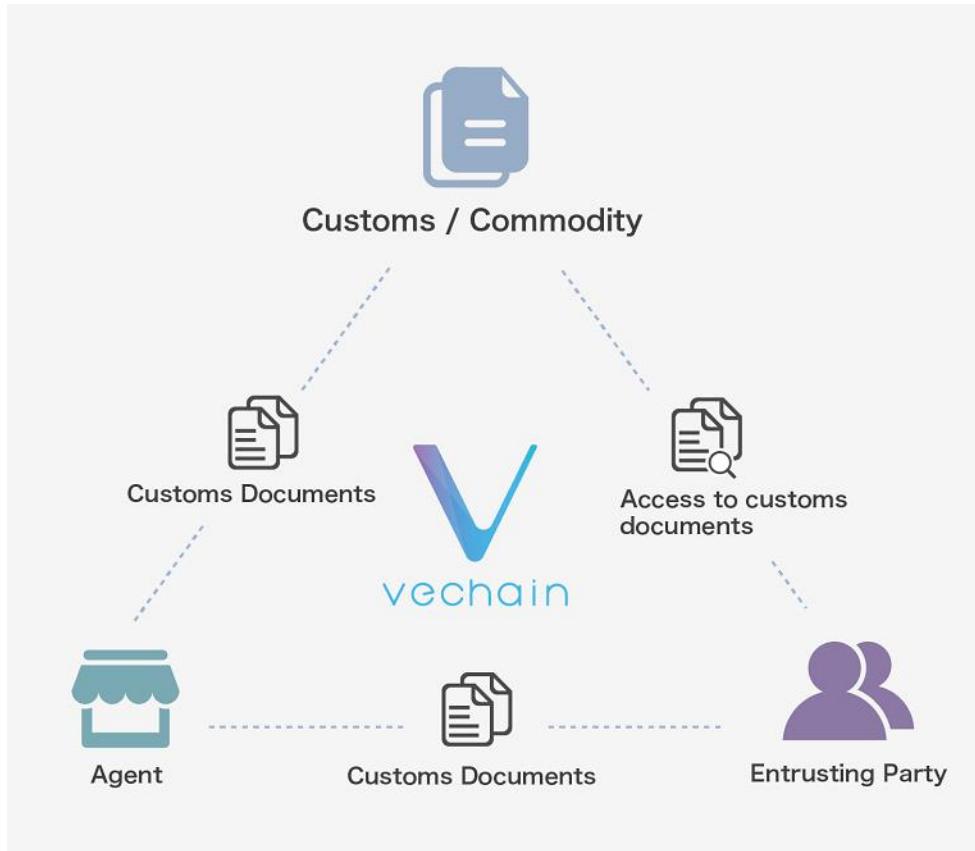


Figure-3.6-2 Blockchain Government Application

The Blockchain technology has shown significant meaning to the government. It represents the quality of government of open, public and transparent about the information. At the same time, as the “coordinator” of the whole commercial environment, government focus more on how Blockchain can improve the efficiency of the resource distribution in different industries.

3.7 This is just the Beginning

In the past two years, team VeChain has faced many big challenges. It's not the technical consensus, but the consensus of how to achieve the business model and change from the traditional business. However, we have been through the worst part already, and we thank so much for the cooperation partners and customers who dared to step into this field with us, so we can develop and verify the practical use of the Blockchain.

VeChain ICO is not just a beginning, because we have the experience from the

past two years. VeChain ICO is also a beginning because in front us, we still have a long way to go in this business environment. The challenge is we still need to invent new business model and promote them in order to complete building the thoroughly connection within the team VeChain, enterprise participant and community.

4. Governance Structure and Management Philosophy

In the beginning of the discussion about ICO and the preparation for the later stage, VeChain team has many heated debates and a long night talk, but we always have a consistent point of view:

"ICO is the beginning of everything, not the end of everything. All the so-called 'successful' ICO, in fact, is only a good starting power. And when the ICO stops, then it's really the time for the beginning of everything. The main theme is always about how to speed up, how to run on the right track, and how to avoid 'dying' during the process. Meanwhile, the team cannot just celebrate because of the ICO initial funding, but this should be regarded as the team bearing the hope of the community/business world and proceeding with great cautions. "

Therefore, maintaining the sustainable development of a team is also a proposition that the VeChain team has been discussing and thinking.

The corporate governance structure from the company system, is used to constrain the enterprise strategy, risk management, operation principle, human resources and legal compliance program.

Although the Blockchain technology utilizes decentralized concept as the starting point and establishes an efficient collaborative community platform, in order to improve the efficiency of collaborative Blockchain community and its operation, the team can still learn from the experience of corporate governance structure. The VeChain is also a framework of "non-traditional" community. In addition to the individual participants, there are more business users from different enterprise would agree to a reasonable corporate governance structure.

Of course, the structure concept cannot be applied mechanically. It is necessary to seek a dynamic balance between community culture and traditional enterprise management culture. This treatment method is combined based on our experience in the Blockchain industry during the past few years with the constant adjusting and optimizing in the future development.

4.1 The Establishment of VeChain Foundation

The VeChain Foundation (will refer as the Foundation hereinafter) is a non - profit entity established in Singapore in July 2017. The foundation will act as VeChain sponsor entity, who committed to support VeChain's development, construction and governance, transparency, advocacy and promotion work, as well as promoting the safety and harmonious development of the community.

The standard Blockchain community aims at a high degree of autonomy or decentralized as goal, allowing community participants to diversify their decision-making advice and usually use "vote" on important matters. However, such behavior is inefficient or unresolved because of the diversity of participants' opinions, which is not conducive to the continuous iteration and evolution of the Blockchain technology.

Moreover, because of the bifurcation behavior of the Blockchain, it causes serious divergence of opinion. The solution of "hard forking" has made people question the idea of "de-centering" of Ethereum" and even "Blockchain". This way of governance is not so much a "democracy" but an "anarchy."

VeChain development team highly recognized Blockchain's "decentralized" as the construction of the essence, while absorbing the essence of the traditional corporate governance structure, and improving the efficient formulation and implementation of the VeChain development strategy. At the same time it also prevent the serious Blockchain design philosophy differences and irreconcilable issue from showing up again.

The VeChain team commissioned a trusted third party organization to assist the team in setting up foundation entities in Singapore and to maintain the day-to-day operations as well as reporting the entity architecture. After establishing the Foundation, it selects the appropriate members of the community to join the functional Committee of the Foundation to participate in the actual management and decision-making.

4.2. Governance Principle

The design objective of the VeChain Foundation governance structure mainly focused on the **sustainability** of VeChain platform, the effectiveness of the strategy formulation, the management effectiveness, the risk control and the efficient operation of the platform economy.

1) The combination of centralized governance and distributed architecture

Although there are arguments advocating that Blockchain is a de-centralized or distributed self-governance community system. We believe that the absolute de-centralized may bring the absolute "fairness" but more likely to be further "inefficient". Therefore, the core idea of the Foundation is to absorb the concept of in the management structure of central governance, including the highest decision-making authority strategic Committee and major issues of the centralized procedure to improve the efficiency of the whole operation of the community.

2) The function of Committee and functional units coexist

The function of Committee and functional units coexist in the Foundation for daily affairs, which will set up permanent functional units, such as R & D department, marketing department, operation department, financial and human resources departments to handle daily affairs.

At the same time, a functional Committee is set up to make decisions on the important functions of the Foundation. Unlike the functional units, functional Committees exist in a virtual architecture where members of the Committee can be from any place of the world and do not have to work full-time. However, it must meet the requirements of the Committee's expert qualifications and be able to undertake to present and make comments when the Committee is required to present. The functional Committee will also set up a regular meeting system to ensure the effective promotion of major decision-making matters.

3) Risk oriented governance principles

The risk management will be the most important element in the process of studying the strategic development of VeChain Foundation. As a computer technology with great on-going revolutionary, the development of Blockchain is still in its infancy, so it is very important to grasp its development trend. The principle of risk management, when making sure the Foundation makes important decisions, takes full account of the risk factors, the possibility and influence of its occurrence, and makes corresponding countermeasures through decision-making. Thus, the development and iteration of the Blockchain is on the right path.

4) Technology and Commerce Coexisting

Any technology that is divorced from commercial applications is often difficult to develop. If the technology lack the practical function, it will stop and even come to the end.

From the creation of the VeChain, it always adhere to the close integration with the business as its purpose. The VeChain Foundation also follows this objective. Even if the Foundation will present as a non-profit organization, but the Foundation wants to get the maximum possible recognition of the business world. To get the business application of revenue, feedback to the Foundation

and the community, while further promoting the Foundation as well as the VeChain's development and upgrade. The VeChain Foundation also takes full account of this principle in the selection of talent and its architecture. It focuses on attracting experts with technical expertise, including industry experts who have a deep understanding of business.

5). Transparency and supervision

Referring to the governance experience of the traditional commercial world, VeChain Foundation also intends to set up special monitoring and reporting channels (Whistle-Blower). Designated by the strategic decision Committee as a window and we welcome the community participants to join in the management, supervision and the operation of Whistle-Blower channel. Those include, but are not limited to, new breakthroughs or recommendations that have significant implications for the Foundation or the Blockchain technologies, community operations issues, crisis information and fraud reporting or fraud, etc.

The Foundation will publish a unified information collection window, while ensuring the privacy of the information collected. At the same time, the Foundation publish periodic reports and irregular news releases in the form of community participation in all parties to disclose the Foundation operation and development of VeChain. Meanwhile, the main contacts of the Foundation management will be fully open, and accept the supervision and liaison of the participants.

4.3 VeChain Governance Model

The organizational structure of VeChain Foundation raised a combination of Specialized Committee and functional departments, which will deal with daily work and special matters. This section will discuss the functions of the functional Committees of the Foundation as well as the functions of the major functional departments.

With the reference to the operation of traditional entities, the Foundation will set up functional Committees, including the strategic decision-making Committee, the technical audit Committee, the remuneration and Nomination Committee and the public relations Committee.

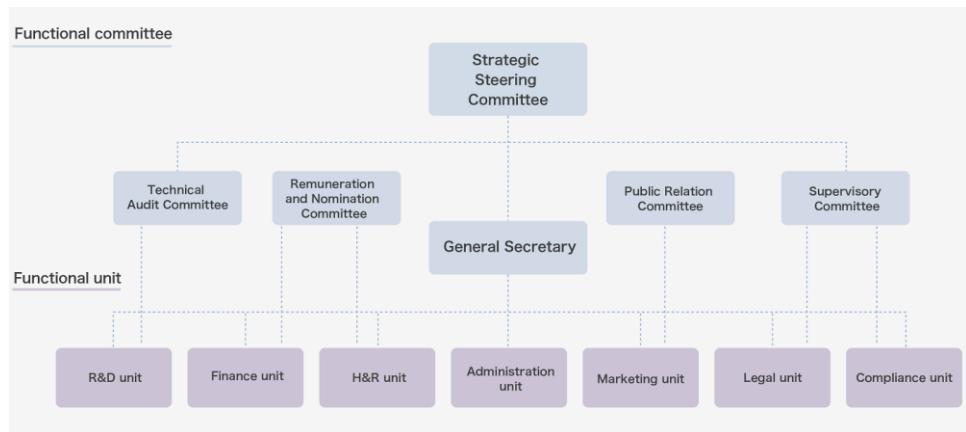


Figure 4.3 VeChain functional Committee structure graphic

4.3.1 Strategic Steering Committee

The strategic decision-making Committee is the highest decision-making Committee. The main objective is to negotiate and solve VeChain's decision making matters faced in community development include, but are not limited to:

- 1) Modifying governance structure of the Foundation.
- 2) The formation and rotation resolution of the policy-making Committee.
- 3) The appointment and rotation resolution of the Secretary General of the Foundation.
- 4) The appointment and dismiss of the chief executive and the head of each functional Committee.
- 5) Foundation review and amendment of the constitution.
- 6) The strategic decision of VeChain development.
- 7) VeChain's core technology changes and upgrades.
- 8) Emergency decisions and crisis management agendas.

The members of the strategic decision Committee and the president of the Foundation will serve for two years and the chairman of the fund shall not be reappointed for more than two sessions.

After the expiration of the decision making Committee term, VeChain will vote 50 community representatives by the consensus mechanism of next generation and then vote for the 7 core personnel of the decision-making Committee. Those elected representatives of the core staff will do emergency decision making, and accept the wage and salary investigation during his tenure, and the public their salary status as well.

These important matters need to be decided by the decision Committee with an open vote. Each member of the policy-making Committee has one vote, and the chairman of the Foundation has two votes. Decisions made by the decision Committee must be approved by more than half of all members of the Committee.

In addition, the person in charge shall convene the decision Committee to hold an interim meeting within 5 working days at the time of the following circumstances:

- ✓ The General Secretary of the Foundation considers when it is necessary.
- ✓ More than 1/3 of the decision Committee members jointly proposed.

The decision Committee meeting shall be attended by the members of the Committee. If members are unable to attend, they may entrust the other members of the Committee in writing. Failing to delegate is deemed to have given up the right to vote at the meeting.

4.3.2 General Secretary

The general secretary is the highest responsible person of VeChain administration. The responsibility is to make guidance and coordinate the daily operation of Foundation, technology development, community maintenance and public relations, as well as connecting various business unit with the governance structure of the functional Committee. The Secretary General will regularly report to the policy-making Committee.

4.3.3 Technical Audit Committee

The audit Committee comprises the core VeChain technology developer, who is responsible for the technology research, development direction of Blockchain, the underlying technology development, open port development and review, technology development and patent examination.

In addition, members of the technical review Committee regularly learn the dynamics and hotspots of the community and industry, communicate with participants in the community, and hold technical seminars on a regular basis.

4.3.4 Remuneration and Nomination Committee

The remuneration and nomination Committee is responsible for determining the selection and appointment of key managers of the Foundation. The Committee shall establish rules of procedure, assess the competence of the management, and authorize the appointment. At the same time, the Committee sets up a compensation system to encourage people who have important contributions to the Foundation.

The remuneration and Nomination Committee regularly reviews the performance of all the Foundation staff, advice on the human resource structure and raise

different incentive measures to attract talented experts.

4.3.5 Public Relation Committee

The public relations Committee is responsible for technically promoting the VeChain within the Committee, business alliance, the establishment and maintenance of the VeChain involved in each alliance party, and publicity regarding community crisis and other social responsibility.

4.3.6 Supervisory Committee

As a highly independent and autonomous form, the supervisory Committee is set up inside the Foundation as an independent risk control for the overall operation of the Foundation. The supervisory Committee conducts day-to-day guidance of the Foundation's legal and compliance departments. At the same time, the Foundation will set up a mechanism for reporting transparency and supervision to receive internal and external reporting issues, take corresponding improvement investigation and treatment, ensure that the Foundation operation is the perfect legal compliance, and continue to move forward in the acceptable level of risk.

The commission reports directly to the Committee on strategic decisions and does not have any conflicts or overlaps with other functions of the Foundation.

4.3.7 Other Functional Department

The Foundation refers to enterprise system framework and sets up day-to-day operations such as human resources, administration, finance, marketing, research and development (or laboratory) units, etc.

The functional departments maintain the normal operation of the VeChain Foundation, and directly deal with the relevant parties in the commercial society, such as enterprise customers, suppliers, regulators and the three party service organizations.

4.4 VeChain Human Resource Management

VeChain is committed to creating the world's most influential open source community ecology. To ensure the smooth development of the technology and the continuity of the Foundation operation, the Foundation will focus on recruiting excellent technology developers and managers with deep understandings of the business.

Talent Recruitment

Based on the characteristics of "Blockchain without borders", the Foundation

welcome talented people from all over the world to join the Foundation. In addition to the individual posts that must be recruited locally (e.g., logistics managers), recruitment is not limited to the place of work or the form of work. VeChain Foundation will, at the same time, follow the best practices in human resource management, develop appropriate human resources plans, recruitment procedures and review procedures to ensure that Foundations attract the right people.

As an open source community, VeChain will not only recruit full-time developers, but also employ well-known industry technical adviser. Relevant hiring and salary payment is required for discussion and decision, and signed the terms of cooperation by remuneration and Nomination Committee.

Performance Appraisal

VeChain will do the performance appraisal based on commercial company's best practice that comprehensively consider technology development, business expansion effect, economic operation, fund risk control management etc. The performance appraisal award will be submitted to the remuneration and Nomination Committee and the Strategic Decision Committee for review, and an optimization plan shall be worked out.

4.5 Risk Assessment and Decision Making Mechanism of VeChain Foundation

As an innovative technology, Blockchain is not only a disruptive breakthrough in computer core technology, but also a challenge to the traditional commercial society. Therefore, the importance of risk management system is self-evident. The VeChain Foundation is committed to build a risk oriented sustainable chain of block communities. It will continue to operate risk management of the foundation which includes the establishment of risk system, risk assessment, risk response and a series of activities.

For major risks, the strategic decision Committee will discuss and make decisions. It will classify risks based on event characteristics, such as event impact, extent of impact, probability of tokens and probability of occurrence, and decisions based on priority. For priority events, the relevant Committees of the foundation shall be organized as soon as possible.

4.6 VeChain Foundation Economy

In its economic operations, the Foundation promotes the following major principles:

- 1) Take the nonprofit as the main principle, and give back to the community;
- 2) Sustainable development
- 3) Collaboration and sharing of resources

Financially, the Foundation will seek the financial balance between expanding and community development. In addition to the initial funding received during the ICO, the Foundation will be able to obtain digital asset income through community eco operations. Under the arrangement of the third party trust institution, it will be transparent to distribute all the benefits to all operations and community development.

The Foundation will set up a full-time financial management team to maintain its financial and digital assets. The financial management team reports directly to the strategic decision Committee, and regularly prepare the financial reports and disclosures of the Foundation.

4.6.1 Funding Sources

The main income of the Foundation can be divided into two areas:

- 1) Non-operating income comprising the initial ICO's funds and the return on digital assets.
- 2) Regular operating income, including R & D, product sales, patent transfer or licensing, academic exchange and contribution, etc.

The following is a detailed description of the main sources of income:

- a. ICO initial startup funds.

VeChain tokens totaling 1 billion VeChain Token(VET). The allocation plan is as follows:

Ratio	Distribution Plan	Details
41%	VeChain Token(VET) crowdsale	The income of VeChain Token(VET) crowdsale will be used to VeChain Foundation operation, including development, marketing, finance and legal advisory.
9%	Private investor	Private investors are very influential in the community, and they will help a lot in technology and business development.
23%	Enterprise investors	Enterprise investor refers to an enterprise in VeChain distributed business ecosystem or a service provider for these corporate customers or end users; these enterprise investors will use the future VeChain Token(VET) as a key development target in their business activities.
5%	Cofounder, development team	To be distributed to the founders and development team of the VeChain Token(VET) as their rewards.

Ratio	Distribution Plan	Details
12%	Continuous operation and technological development	To be reserved for various operating costs and development of the VeChain.
10%	Business development case	To Choose the suitable industry, using VeChain technology to the strategic deployment of the industry, project support and tokens replacement.

- b. Digital asset investment. During continuous operation, the Foundation will allocate about 5% to 10% of the funds or digital assets to invest in the Blockchain industry, such as start-ups and incubators, angel investment in emerging scientific and technological investment.
- c. In the process of building the ecosystem, VeChain will serve as an underlying architectures provider of VeChain and receive a certain amount of digital assets or funds. For example, community participants, enterprises and other VeChain Token(VET) purchasers for GAS, as well as the Foundation. It will provide technical sharing and licensing gains. For this part of the proceeds, the Foundation will continue to invest in the community, form a community constantly expanding, and increase the positive cycle of influence.

4.6.2 Fund Budgeting

As mentioned above, the Foundation's funds spending mainly includes day-to-day operations, technology development, business development and reinvestment. The main categories are shown in the following table:

Classification	Percentage	Content
Technology Development	50%	It mainly includes reward for initial team, recruitment of experts and developers, technical patent and protection of intellectual property rights
Business Development	35%	VeChain business development and training, technical exchange and sharing, periodical publication, alliance establishment or participation, etc.
Reinvestment	10%	Blockchain, new technology and new team investment or absorption
Daily Operation	5%	Foundation daily logistics management, transportation and office, financial and reporting needs, etc.

See the foundation's initial forecast for the next four years of its operations:

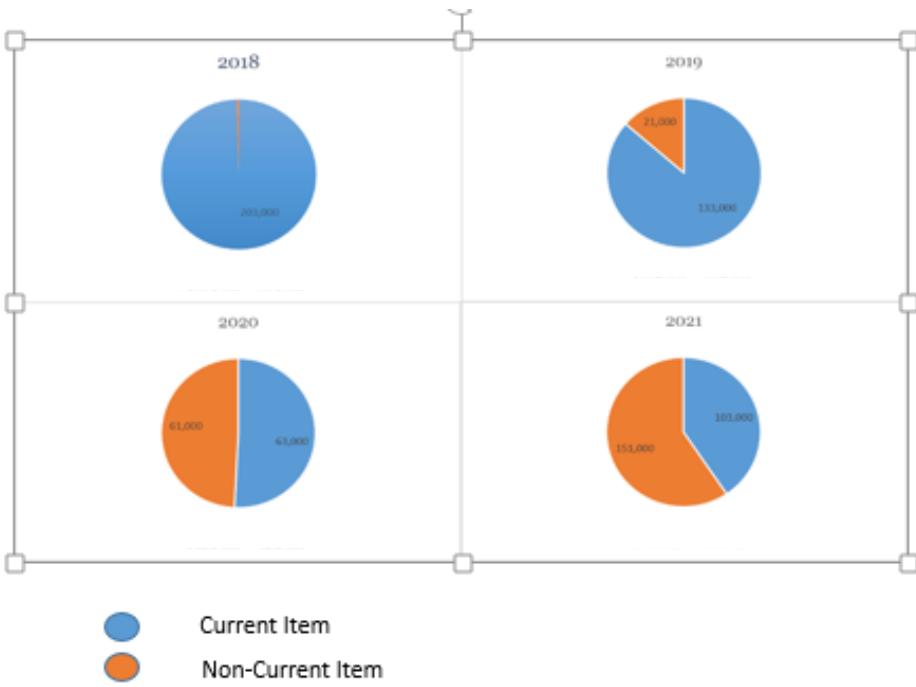


Figure 4.6.2-1 VeChain Foundation's 4 annual revenue forecast (000 RMB)

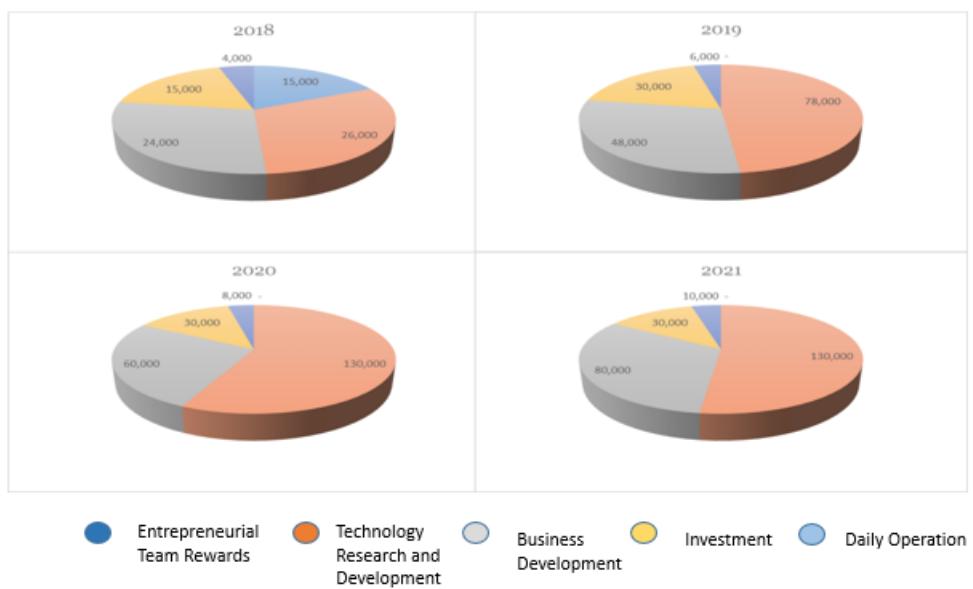


Figure 4.6.2-1 VeChain Foundation 4 year cost forecast (000 RMB)

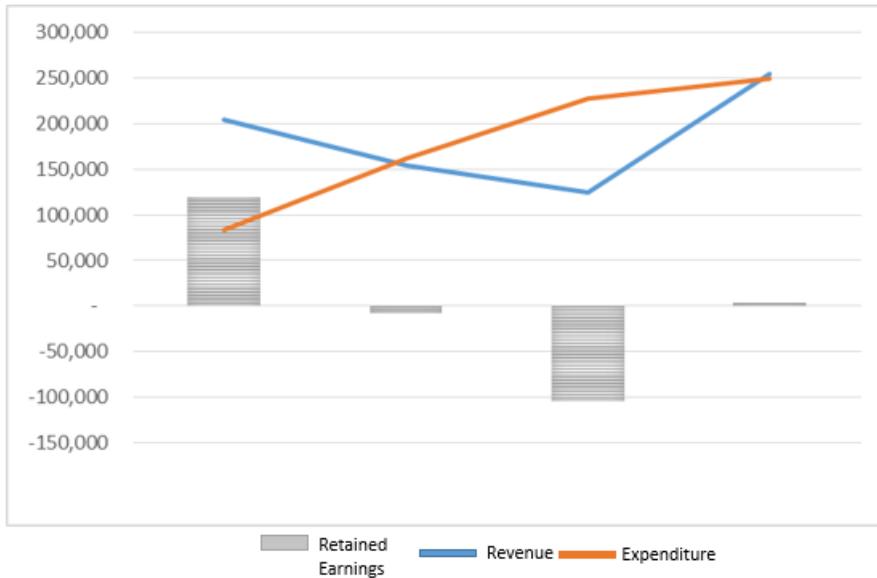


Figure 4.6.2-1 the VeChain Foundation's 4 year retained earnings forecast (000 RMB)

To sum up, the VeChain Foundation is expected to obtain start-up funds through ICO activities, which took about 3-4 years to achieve:

- 1) **Foundation scale and influence continues to grow.** This includes headcounts increasing to around 100. The Foundation attracted the business world continuously joining with than 150 billion yuan of goods in the VeChain flow.
- 2) **Foundation completes the self-circulation.** The Foundation relies on initial ICO start-up funds to get commercial value from the community and feed back to the community. The Foundation guarantees that the gains will be balanced with the expenditure.
- 3) **Focusing on R & D and commercial promotion.** According to the Foundation and VeChain concept, Foundation has always attached importance to the Blockchain based research and development, and business promotion and expand the influence. Most of the annual expenditure will focus on these two aspects.
- 4) **Adhere to the nonprofit principle.** The Foundation promises not to distribute profits, nor does it call dividends". Foundation operating income, in addition to the basic expenditure of the foundation, will all be put into the expansion of the community, to promote the community growing.

4.6.3 Fund Use Restriction

The use of VeChain assets is in line with the principles of openness and transparency. According to the principle of distribution and budget, VeChain will set up a separate account and digital asset wallet address used by depository

institutions to digital assets supervision and regularly share to the community. The principle of the use of revenue from public sale:

- ✓ Exceeding the value of 1 million yuan (or equivalent digital assets) requires approval by the head of the financial unit and the Secretary General.
- ✓ Over 5 million yuan (or equivalent digital assets) will need to be approved by the policy-making Committee.

4.6.4 Financial Planning and Implementation Reports

Each quarter, the financial and personnel management Committee prepare the financial planning, and summarize the last quarter financial performance. The formation of financial reports will be submitted to the decision-making Committee for approval.

4.6.5 Digital Asset Management

The digital assets belonging to the Foundation are appointed by the strategic decision Committee, and the full-time financial personnel are responsible for the arrangement. Digital assets and transaction currency are arranged independently and timely financial accounting. Following the best practices of financial control, the Foundation adopts multiple signatures to ensure the safety and accuracy of the assets. All the collected coins will be the timely transfer to digital assets and digital wallet. Foundation assets are not deposited in individual accounts.

- **Digital wallet management**

Based on the principle of independence, VeChain Foundation's wallet adopts 4/7 multi signature. Added Signature is subject to the approval of the strategic decision Committee. Large tokens are cold saved, and small tokens use multiple signatures.

- **Disclosure matters**

Each year, the Foundation will inform the community of chain development, operations, business promotion and the Foundation's operations. For the financial situation of the Foundation, the financial statements will be performed quarterly, and the work of the annual audit will be disclosed as well.

The Foundation establishes a public relations Committee, which serves as an external window for regular and irregular meetings and releases important information to the public.

4.7 Legal Compliance Matters and Other Matters

- **Legal affairs**

VeChain team commissioned a trusted third party organization to set up a Foundation entity in Singapore. All operations are subject to local laws, regulations and regulatory requirements. If there is a need to seek legal advice, it needs to be confirmed by local counsel.

- **Exemption clause**

VeChain Foundation insists the nonprofit nature of the unit's operations. Whether or not to obtain only chain tokens, Users who participate in the only chain community, can hold token or give up token rights. Holding tokens simultaneously means the holder's own rights to consume and use smart contracts on the Blockchain platform. Buyers should understand that within the scope of the law, VeChain foundation does not make any express or implied warranties and benefits. In addition, buyers should understand that there is no refund or refund after purchasing only chain tokens.

- **Settlement of dispute clause**

When a dispute arises, the parties concerned shall settle it by consultation in accordance with the agreement. If the settlement cannot be solved by negotiation, it can be settled by law

5. Introduction of the Team and Team Member

Team VeChain is a pure internationalized team. The team member is from different industries and countries, but they follows the same dream. The composition of the team is well balanced. Business, technology, operation and support are all important.



Sunny Lu , Project Leader

Sunny was graduated from Shanghai Jiao Tong University, majored in Electronics and Communication Engineering. He has been served as IT Executive in Fortune 500 companies over 13 years, former CIO of LV China.

He started VeChain project in 2015, and committed to Blockchain technology and business implementation.



Richard Fu, PR & Marketing Director

Richard has over 20 years' working experience in multi-national enterprises such as Shangri-la Group and LVHM specializing in sales and marketing.

He joined VeChain as director of PR and marketing



Chin Qian, Channel & Sales Director

Chin worked for HP from 2004 to 2016 and accumulated rich experience in marketing and project management.

He joined VeChain in 2017 as director of business partner recruitment and management.



Jay Zhang, Finance Director

Jay has worked for PwC and Deloitte as senior manager over 14 years.

He joined VeChain in 2015 as the leader of Blockchain governance framework design and digital assets management framework establishment.



Scott Brsbin, General Counsel

Scott is a well-known lawyer from the United States. His clients include Rolling Stones and lead singer Mick Jagger, Disney, MGM and so on.

He graduated from the University of California, Los Angeles in 1978, he joined the MSK law firm, and since 1989 began as a legal partner. In the company's legal affairs and patent maintenance, he has an absolute authority on the experience.

Scott joined VeChain in 2016 and worked for VeChain's legal security, organizational structure and property

escort.



Jerome Grilleres, Business Development

Jerome holds an MBA from London Business School and a MSc in Computer Science. He is from Barclays France and has 8 years' experience in Business Strategy and Development in Retail Banking and 6 years in Developing Real Time trading application in Investment Banks. Jerome joined VeChain in 2017 as Business Director of Europe.



Jianliang Gu, Technical Director

Jianliang was graduated from Shanghai University with master degree majored in Cybernetics. He was working in TCL communication technology as Technical Director. He has more than 16 years' experience in both hardware and software of embedded system development and management.

He joined VeChain in 2017 and commit to marry IoT and Blockchain



Peter Zhou, R&D director & Scientist

Dr. Peter Zhou obtained his Ph.D degrees in Computer Science from the University of Southampton. He was involved in projects funded by the European Commission and Academy of Finland when working as a postdoctoral researcher at the University of Kent, UK as well as a senior research scientist at the University of Oulu, Finland.

He has had more than ten years of scientific research experiences and published papers in top-tier international journals and conferences.



Bin Qian, Blockchain Director

Bin has over 10 years' experience in Mobile application development industry, specializing in developing Internet applications based real-time communication system. He is definitely P2P network technology expert. He joined VeChain in 2016 and is in charge of the Blockchain development.



Tony Li, Application development manager

Tony's majored in Information Security. He has 5 years' experience developing software and project management. Took part in numerous projects, including financial industry, insurance industry, luxury industry, the automotive industry.

He's interested in Bitcoin and Blockchain technology since 2014, and has two years' experience developing in the Blockchain product development.



Sherry Li, Product Manager

Sherry was graduated from Jiangnan University majored in Information Security. She has over 4 years of experience in application development, project management and product planning, including SAAS service platform, O2O platform and user oriented application.

She joined VeChain in 2016 as Product Manager



Jack Wu, Project Manager

Jack was graduated from St.John's university (New York). He has over 3 years iOS development and project management experiences. Took part in numerous project, including luxury goods industry, government agencies, the automotive industry
He joined VeChain in 2016 as Blockchain Project Manager.



Harvey Shang, DA Facilitator

Harvey was graduated from University of Florida majored in Computer Science. His study focus in distributed systems and advanced data structure.
He joined VeChain in 2016 and continued working in researching digital assets management area.



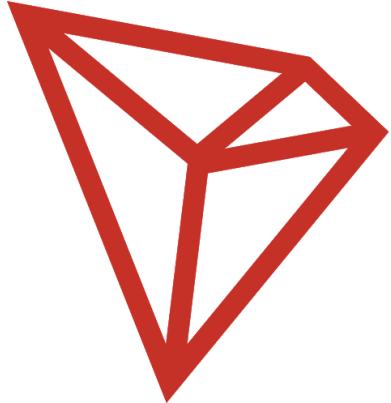
Lingbo Li, Risk Controller

Lingbo was graduated from Chinese Academy of Sciences with master degree in Finance Engineering. She has over 11 years' experience in credit risk management and assets management.
She joined VeChain in 2016 and is responsible for digital assets management and related risk control.



Cissy Chen, HR&Admin Manager

Cissy has over 6 years' experience in human resource management and worked for sub brands of Unilever before join VeChain in 2015.
She is in charge of human resource management, recruitment, staff training, compensation and other related strategies and policies establishment.



TRON

Advanced Decentralized Blockchain Platform

Whitepaper Version: 2.0

TRON Protocol Version: 3.2

TRON Foundation

December 10th, 2018, San Francisco

1. Introduction	4
1.1 Vision	4
1.2 Background	4
1.3 History	5
1.4 Terminology	6
Address/Wallet	6
ABI	6
API	6
Asset	6
Bandwidth Points (BP)	6
Block	6
Block Reward	6
Block Header	6
Cold Wallet	7
DApp	7
gRPC	7
Hot Wallet	7
JDK	7
KhaosDB	7
LevelDB	7
Merkle Root	7
Public Testnet (Shasta)	8
RPC	8
Scalability	8
SUN	8
Throughput	8
Timestamp	8
TKC	8
TRC-10	8
TRX	8
2. Architecture	9
2.1 Core	10
2.2 Storage	10
2.2.1 Blockchain Storage	10
2.2.2 State Storage	10

2.3 Application	10
2.4 Protocol	11
2.4.1 Protocol Buffers	11
2.4.2 HTTP	11
2.5 TRON Virtual Machine (TVM)	11
2.6 Decentralized Exchange (DEX)	11
2.7 Implementation	12
3. Consensus	13
3.1 Delegated Proof of Stake (DPoS)	13
4. Account	16
4.1 Types	16
4.2 Creation	16
4.3 Structure	16
5. Block	18
5.1 Block Header	18
5.1.1 Raw Data	18
5.1.2 Witness Signature	19
5.1.3 Block ID	19
5.2 Transaction	19
5.2.1 Signing	19
5.2.2 Bandwidth Model	19
5.2.3 Fee	20
5.2.4 Transaction as Proof of Stake (TaPoS)	20
5.2.5 Transaction Confirmation	21
5.2.6 Structure	21
6. TRON Virtual Machine (TVM)	23
6.1 Introduction	23
6.2 Workflow	23
6.3 Performance	25
6.3.1 Lightweight Architecture	25
6.3.2 Robust	25
6.3.3 High Compatibility	25
6.3.4 Low Cost	25
7. Smart Contract	26
7.1 Introduction	26
7.2 Energy Model	26
7.3 Deployment	27

7.4 Trigger Function	27
7.5 TRON Solidity	27
8. Token	28
8.1 TRC-10 Token	28
8.2 TRC-20 Token	28
8.3 Beyond	29
9. Governance	30
9.1 Super Representative	30
9.1.1 General	30
9.1.2 Election	30
9.1.3 Reward	30
a. Vote Reward	30
b. Block Reward	31
c. Reward Calculation	31
9.2 Committee	32
9.2.1 General	32
9.2.2 Dynamic Network Parameters	32
9.2.3 Create Proposal	36
9.2.4 Vote Proposal	36
9.2.5 Cancel Proposal	36
9.3 Structure	36
10. DApp Development	37
10.1 APIs	37
10.2 Networks	37
10.3 Tools	37
10.4 Resources	37
11. Conclusion	39

1. Introduction

1.1 Vision

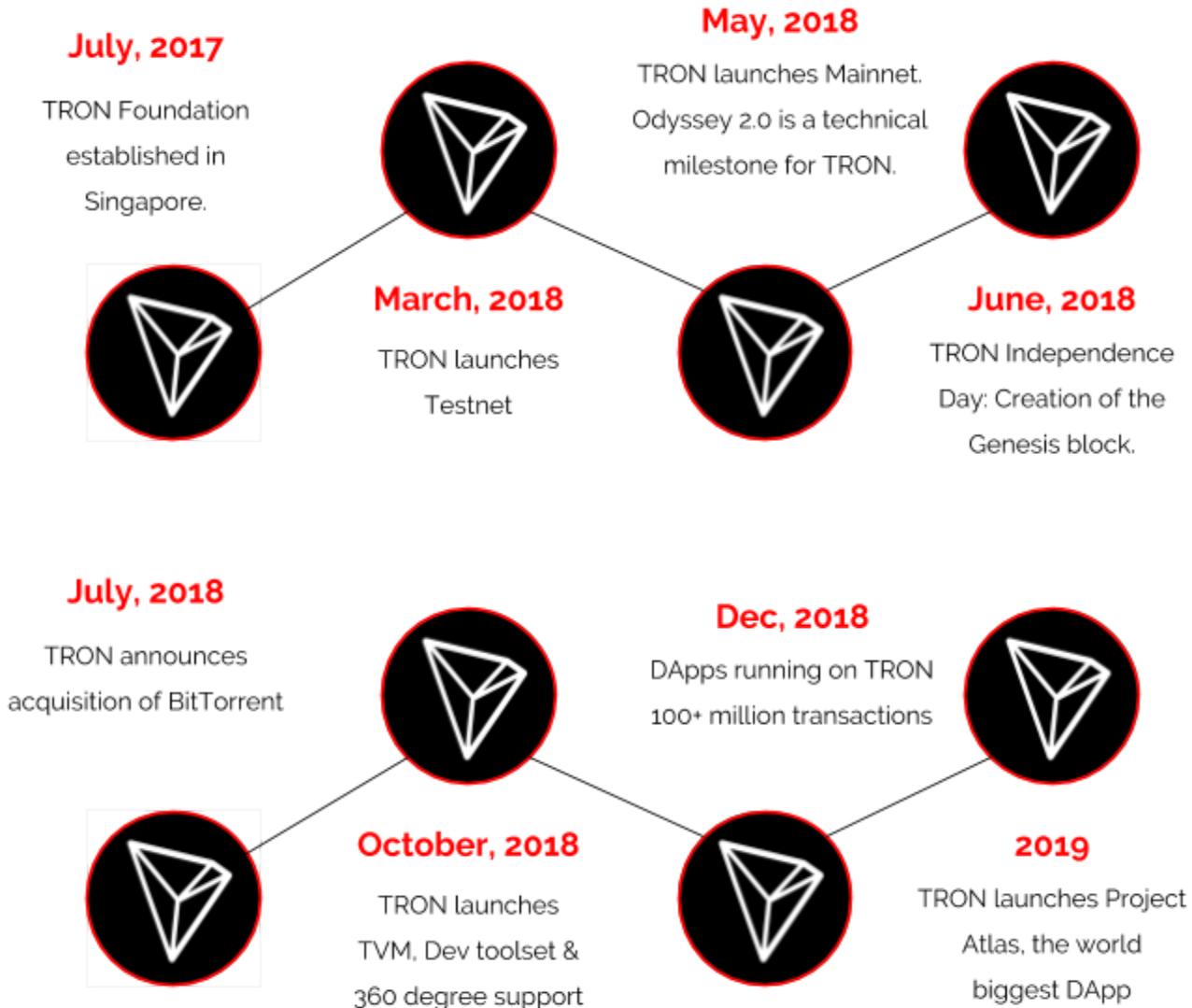
TRON is an ambitious project dedicated to the establishment of a truly decentralized Internet and its infrastructure. The TRON Protocol, one of the largest blockchain-based operating systems in the world, offers public blockchain support of high throughput, high scalability, and high availability for all Decentralized Applications (DApps) in the TRON ecosystem. The July 2018 acquisition of BitTorrent further cemented TRON's leadership in pursuing a decentralized ecosystem.

1.2 Background

The introduction of Bitcoin in 2009 revolutionized society's perception of the traditional financial system in the wake of the Great Recession (2007-2008). As centralized hedge funds and banks collapsed from speculation in opaque financial derivatives, blockchain technology provided a transparent universal ledger from which anybody could glean transaction information. The transactions were cryptographically secured using a Proof of Work (PoW) consensus mechanism, thus preventing double spend issues.

In late 2013, the Ethereum white paper proposed a network in which smart contracts and a Turing-complete Ethereum Virtual Machine (EVM) would allow developers to interact with the network through DApps. However, as transaction volumes in Bitcoin and Ethereum peaked in 2017, it was apparent from the low transaction throughput times and high transaction fees that cryptocurrencies like Bitcoin and Ethereum in their existing state were not scalable for widespread adoption. Thus, TRON was founded and envisioned as an innovative solution to these pressing scalability challenges.

1.3 History



The TRON Foundation was established in July 2017 in Singapore. In December 2017, TRON had launched its open source protocol. The Testnet, Blockchain Explorer, and Web Wallet were all launched by March 2018. TRON Mainnet launched shortly afterward in May 2018, marking the Odyssey 2.0 release as a technical milestone. In June 2018, TRON declared its independence with the creation of the Genesis block, along with the July 2018 acquisition of BitTorrent. In October 2018, TRON launched the TRON Virtual Machine (TVM), a complete developers' toolset, and 360 support system. The TRON roadmap involves combining BitTorrent's 100 million users with the TRON network via Project Atlas, as well as fostering the developer community to launch exciting new DApps on the TRON network¹.

¹ V1.0 is available at https://tron.network/static/doc/white_paper_v_1_0.pdf

1.4 Terminology

Address/Wallet

An address or wallet consisting of account credentials on the TRON network are generated by a key pair, which consists of a private key and a public key, the latter being derived from the former through an algorithm. The public key is usually used for session key encryption, signature verification, and encrypting data that could be decrypted by a corresponding private key.

ABI

An application binary interface (ABI) is an interface between two binary program modules; usually one of these modules is a library or an operating system facility, and the other is a user run program.

API

An application programming interface (API) is mainly used for user clients development. With API support, token issuance platforms can also be designed by developers themselves.

Asset

In TRON's documents, asset is the same as token, which is also denoted as TRC-10 token.

Bandwidth Points (BP)

To keep the network operating smoothly, TRON network transactions use BP as fuel. Each account gets 5000 free daily BP and more can be obtained by freezing TRX for BP. Both TRX and TRC-10 token transfers are normal transactions costing BP. Smart contract deployment and execution transactions consume both BP and Energy.

Block

Blocks contain the digital records of transactions. A complete block consists of the magic number, block size, block header, transaction counter, and transaction data.

Block Reward

Block production rewards are sent to a sub-account (address/wallet). Super Representatives can claim their rewards on Tronscan or through the API directly.

Block Header

A block header is part of a block. TRON block headers contain the previous block's hash, the Merkle root, timestamp, version, and witness address.

Cold Wallet

Cold wallet, also known as offline wallet, keeps the private key completely disconnected from any network. Cold wallets are usually installed on "cold" devices (e.g. computers or mobile phones staying offline) to ensure the security of TRX private key.

DApp

Decentralized Application is an App that operates without a centrally trusted party. An application that enables direct interaction/agreements/communication between end users and/or resources without a middleman.

gRPC

gRPC² (gRPC Remote Procedure Calls) is an open source remote procedure call (RPC) system initially developed at Google. It uses HTTP/2 for transport, Protocol Buffers as the interface description language, and provides features such as authentication, bidirectional streaming and flow control, blocking or nonblocking bindings, and cancellation and timeouts. It generates cross-platform client and server bindings for many languages. Most common usage scenarios include connecting services in microservices style architecture and connecting mobile devices, and browser clients to backend services.

Hot Wallet

Hot wallet, also known as online wallet, allows user's private key to be used online, thus it could be susceptible to potential vulnerabilities or interception by malicious actors.

JDK

Java Development Kit is the Java SDK used for Java applications. It is the core of Java development, comprising the Java application environment (JVM+Java class library) and Java tools.

KhaosDB

TRON has a KhaosDB in the full-node memory that can store all the newly-forked chains generated within a certain period of time and supports witnesses to switch from their own active chain swiftly into a new main chain. See 2.2.2 State Storage for more details.

LevelDB

LevelDB was initially adopted with the primary goal to meet the requirements of fast R/W and rapid development. After launching the Mainnet, TRON upgraded its database to an entirely customized one catered to its very own needs. See 2.2.1 Blockchain Storage for more details.

Merkle Root

A Merkle root is the hash of all hashes of all transactions included as part of a block in a blockchain network. See 3.1 Delegated Proof of Stake (DPoS) for more details.

² <https://en.wikipedia.org/wiki/GRPC>

Public Testnet (Shasta)

A version of the network running in a single-node configuration. Developers can connect and test features without worrying about the economic loss. Testnet tokens have no value and anyone can request more from the public faucet.

RPC³

In distributed computing, a remote procedure call (RPC) is when a computer program causes a procedure (subroutine) to execute in a different address space (commonly on another computer on a shared network), which is coded as if it were a normal (local) procedure call, without the programmer explicitly coding the details for the remote interaction.

Scalability

Scalability is a feature of the TRON Protocol. It is the capability of a system, network, or process to handle a growing amount of work or its potential to be enlarged to accommodate that growth.

SUN

SUN replaced drop as the smallest unit of TRX. 1 TRX = 1,000,000 SUN.

Throughput

High throughput is a feature of TRON Mainnet. It is measured in Transactions Per Second (TPS), namely the maximum transaction capacity in one second.

Timestamp

The approximate time of block production is recorded as Unix timestamp, which is the number of milliseconds that have elapsed since 00:00:00 01 Jan 1970 UTC.

TKC

Token configuration.

TRC-10

A standard of crypto token on TRON platform. Certain rules and interfaces are required to follow when holding an initial coin offering on TRON blockchain.

TRX

TRX stands for Tronix, which is the official cryptocurrency of TRON.

³ https://en.wikipedia.org/wiki/Remote_procedure_call

2. Architecture

TRON adopts a 3-layer architecture divided into Storage Layer, Core Layer, and Application Layer. The TRON protocol adheres to Google Protobuf, which intrinsically supports multi-language extension.

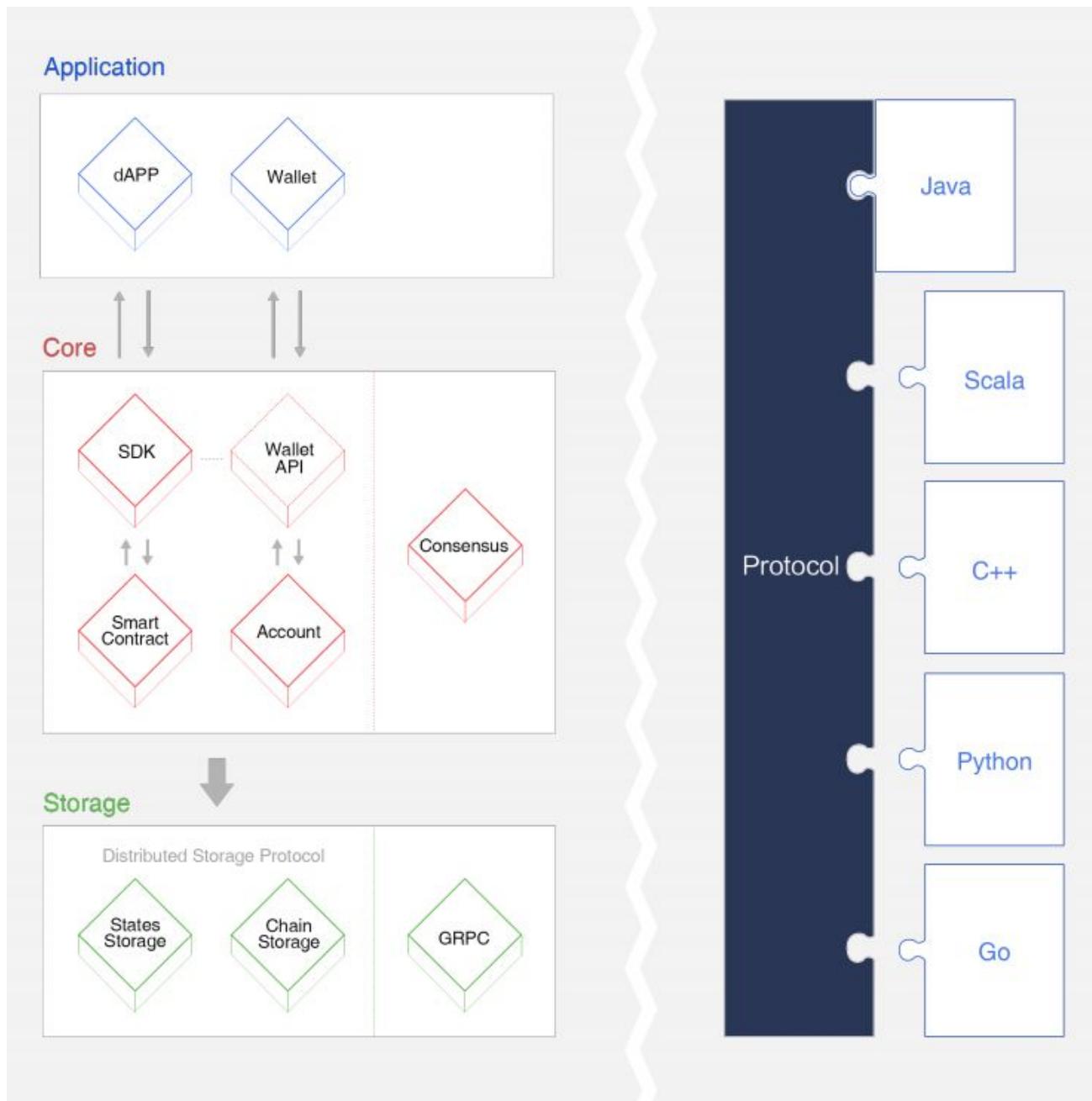


Figure 1: TRON 3-layer Architecture

2.1 Core

There are several modules in the core layer, including smart contracts, account management, and consensus. A stack-based virtual machine is implemented on TRON and an optimized instruction set is used. In order to better support DApp developers, Solidity⁴ was chosen as the smart contract language, followed by future support of other advanced languages. In addition, TRON's consensus mechanism is based on Delegated Proof of Stake (DPoS) and many innovations were made in order to meet its unique requirements.

2.2 Storage

TRON designed a unique distributed storage protocol consisting of Block Storage and State Storage. The notion of a graph database was introduced into the design of the storage layer to better meet the need for diversified data storage in the real world.

2.2.1 Blockchain Storage

TRON blockchain storage chooses to use LevelDB, which is developed by Google and proven successful with many companies and projects. It has high performance and supports arbitrary byte arrays as both keys and values, singular get, put and delete, batched put and delete, bi-directional iterators, and simple compression using the very fast Snappy algorithm.

2.2.2 State Storage

TRON has a KhaosDB in the full-node memory that can store all the newly forked chains generated within a certain period of time and supports witnesses to switch from their own active chain swiftly into a new main chain. It can also protect blockchain storage by making it more stable from being terminating abnormally in an intermediate state.

2.3 Application

Developers can create a diverse range of DApps and customized wallets on TRON. Since TRON enables smart contracts to be deployed and executed, the opportunities of utility applications are unlimited.

⁴ Solidity official documentation: <https://solidity.readthedocs.io/>

2.4 Protocol

TRON protocol adheres to Google Protocol Buffers⁵, which is a language-neutral, platform-neutral, and extensible way of serializing structured data for use in communications protocols, data storage, and more.

2.4.1 Protocol Buffers

Protocol Buffers (Protobuf) is a flexible, efficient, automated mechanism for serializing structured data, similar to JSON or XML, but much smaller, faster and simpler.

Protobuf (.proto) definitions can be used to generate code for C++, Java, C#, Python, Ruby, Golang, and Objective-C languages through the official code generators. Various third-party implementations are also available for many other languages. Protobuf eases development for clients by unifying the API definitions and also optimizing data transfers. Clients can take the API .proto from TRON's protocol repository and integrate through the automatically-generated code libraries.

As a comparison, Protocol Buffers is 3 to 10 times smaller and 20 to 100 times faster than XML, with less ambiguous syntax. Protobuf generates data access classes that are easier to use programmatically.

2.4.2 HTTP

TRON Protocol provides a RESTful HTTP API alternative to the Protobuf API. They share the same interface but the HTTP API can be readily used in javascript clients.

2.5 TRON Virtual Machine (TVM)

The TVM is a lightweight, Turing complete virtual machine developed for TRON's ecosystem. The TVM connects seamlessly with the existing development ecosystem to provide millions of global developers with a custom-built blockchain system that is efficient, convenient, stable, secure, and scalable.

2.6 Decentralized Exchange (DEX)

⁵ Google Protocol Buffers official documentation: <https://developers.google.com/protocol-buffers/>

The TRON network natively supports decentralized exchange functions. A decentralized exchange consists of multiple trading pairs. A trading pair (notation “Exchange”) is an Exchange Market between TRC-10 tokens, or between a TRC-10 token and TRX. Any account can create a trading pair between any tokens, even if the same pair already exists on the TRON network. Trading and price fluctuations of the trading pairs follow the Bancor Protocol⁶. The TRON network stipulates that the weights of the two tokens in all trading pairs are equal, so the ratio of their balances is the price between them. For example, consider a trading pair containing two tokens, ABC and DEF. ABC has a balance of 10 million and DEF has a balance of 1 million. Since their weights are equal, $10 \text{ ABC} = 1 \text{ DEF}$. This means that the ratio of ABC to DEF is 10 ABC per DEF.

2.7 Implementation

The TRON blockchain code is implemented in Java and was originally a fork from EthereumJ.

⁶ Bancor Protocol official website: <https://about.bancor.network/protocol/>

3. Consensus

3.1 Delegated Proof of Stake (DPoS)

The earliest consensus mechanism is the Proof of Work (PoW) consensus mechanism. This protocol is currently implemented in Bitcoin⁷ and Ethereum⁸. In PoW systems, transactions broadcast through the network are grouped together into nascent blocks for miner confirmation. The confirmation process involves hashing transactions using cryptographic hashing algorithms until a merkle root has been reached, creating a merkle tree:

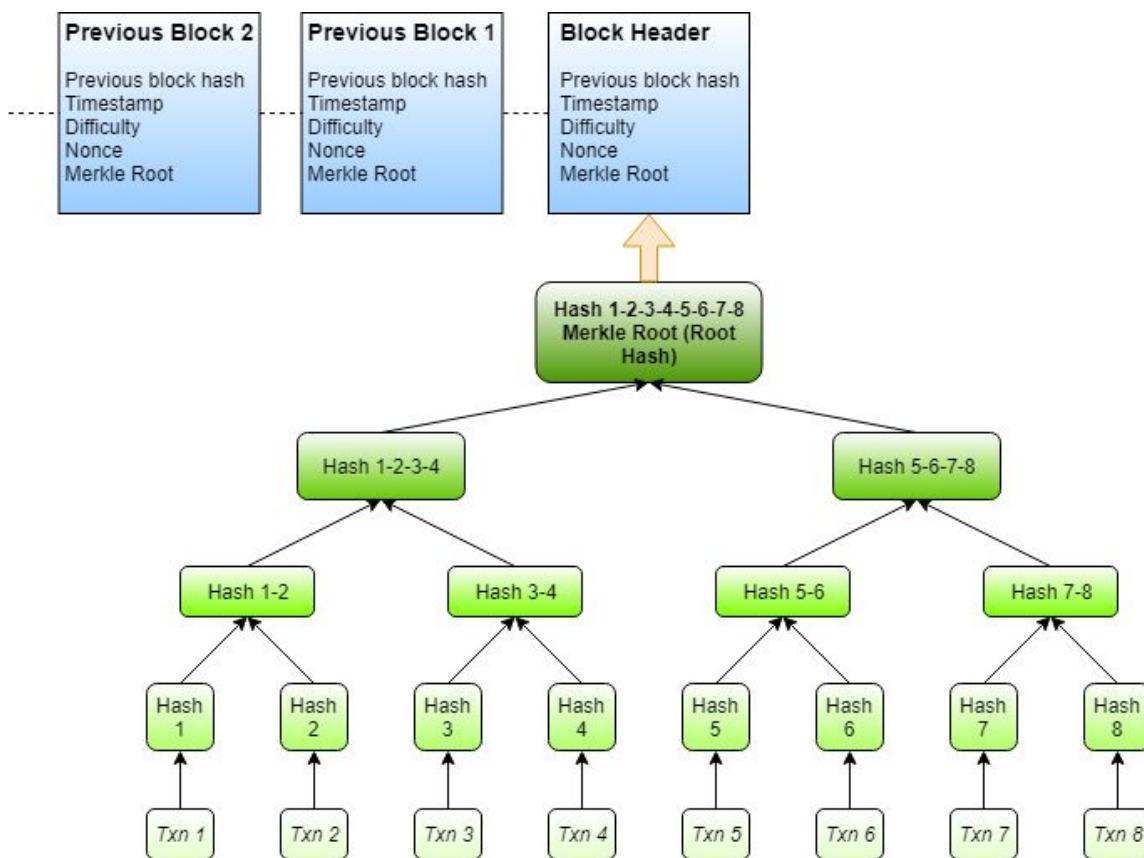


Figure 2: 8 TRX transactions are hashed into the merkle root. This merkle root is then included in the block header, which is attached to the previously confirmed blocks to form a blockchain. This allows for easy and transparent tracking of transactions, timestamps, and other related information.

⁷ Bitcoin whitepaper: <https://bitcoin.org/bitcoin.pdf>

⁸ Ethereum whitepaper: <https://github.com/ethereum/wiki/wiki/White-Paper>

Cryptographic hashing algorithms are useful in network attack prevention because they possess several properties⁹:

- **Input/Output length size** - The algorithm can pass in an input of any length in size, and outputs a fixed length hash value.
- **Efficiency** - The algorithm is relatively easy and fast to compute.
- **Preimage resistance** - For a given output z , it is impossible to find any input x such that $h(x) = z$. In other words, the hashing algorithm $h(x)$ is a one-way function in which only the output can be found, given an input. The reverse is not possible.
- **Collision resistance** - It is computationally infeasible to find any pairs $x_1 \neq x_2$ such that $h(x_1) = h(x_2)$. In other words, the probability of finding two different inputs hashing to the same output is extremely low. This property also implies *second preimage resistance*.
- **Second preimage resistance** - Given x_1 , and thus $h(x_1)$, it is computationally infeasible to find any x_2 such that $h(x_1) = h(x_2)$. While this property is similar to *collision resistance*, the property differs in that it is saying an attacker with a given x , will find it computationally infeasible to find any x_2 hashing to the same output.
- **Deterministic** - maps each input to one and only one output.
- **Avalanche effect** - a small change in the input results in an entirely different output.

These properties give the cryptocurrency network its intrinsic value by ensuring attacks do not compromise the network. When miners confirm a block, they are rewarded tokens as a built-in incentive for network participation. However, as the global cryptocurrency market capitalization steadily increased, the miners became centralized and focused their computing resources on hoarding tokens as assets, rather than for network participation purposes. CPU miners gave way to GPUs, which in turn gave way to powerful ASICs. In one notable study, the total power consumption of Bitcoin mining has been estimated to be as high as 3 GW¹⁰, comparable to Ireland's power consumption. This same study projected total power consumption to reach 8 GW in the near future.

To solve the energy waste issue, the Proof of Stake (PoS) consensus mechanism was proposed by many new networks. In PoS networks, token holders lock their token balances to become block validators. The validators take turns proposing and voting on the next block. However, the problem with standard PoS is that validator influence correlates directly to the amount of tokens locked up. This results in parties hoarding large amounts of the network's base currency wielding undue influence in the network ecosystem.

The TRON consensus mechanism uses an innovative Delegated Proof of Stake system in which 27 Super Representatives (SRs) produce blocks for the network. Every 6 hours, TRX account holders who freeze their accounts can vote for a selection of SR candidates, with the top 27 candidates deemed the SRs. Voters may choose SRs based on criteria such as projects sponsored by SRs to

⁹ PAAR, C., PELZL, J., *Understanding Cryptography: A Textbook for Students and Practitioners*, 2010 ed. Springer-Verlag Berlin Heidelberg, 2010.

¹⁰ <https://www.sciencedirect.com/science/article/pii/S2542435118301776>

increase TRX adoption, and rewards distributed to voters. This allows for a more democratized and decentralized ecosystem. SRs' accounts are normal accounts, but their accumulation of votes allows them to produce blocks. With the low throughput rates of Bitcoin and Ethereum due to their PoW consensus mechanism and scalability issues, TRON's DPoS system offers an innovative mechanism resulting in 2000 TPS compared to Bitcoin's 3 TPS and Ethereum's 15 TPS.

The TRON protocol network generates one block every three seconds, with each block awarding 32 TRX to Super Representatives. A total of 336,384,000 TRX will be awarded annually to the 27 SRs. Each time an SR finishes block production, rewards are sent to a sub-account in the super-ledger. SRs can check, but not directly make use of these TRX tokens. A withdrawal can be made by each SR once every 24 hours, transferring the rewards from the sub-account to the specified SR account.

The three types of nodes on the TRON network are Witness Node, Full Node, and Solidity Node. Witness nodes are set up by SRs and are mainly responsible for block production and proposal creation/voting. Full nodes provide APIs and broadcast transactions and blocks. Solidity nodes sync blocks from other Full Nodes and also provide indexable APIs.

4. Account

4.1 Types

The three types of accounts in the TRON network are regular accounts, token accounts, and contract accounts.

1. Regular accounts are used for standard transactions.
2. Token accounts are used for storing TRC-10 tokens.
3. Contract accounts are smart contract accounts created by regular accounts and can be triggered by regular accounts as well.

4.2 Creation

There are three ways to create a TRON account:

1. Create a new account through API
2. Transfer TRX into a new account address
3. Transfer any TRC-10 token into a new account address

An offline key-pair consisting of an address (public key) and a private key, and not recorded by the TRON network, can also be generated. The user address generation algorithm consists of generating a key-pair and then extracting the public key (64-byte byte array representing x, y coordinates). Hash the public key using the SHA3-256 function (the SHA3 protocol adopted is KECCAK-256) and extract the last 20 bytes of the result. Add 41 to the beginning of the byte array and ensure the initial address length is 21 bytes. Hash the address twice using SHA3-256 function and take the first 4 bytes as verification code. Add the verification code to the end of the initial address and obtain the address in base58check format through base58 encoding. An encoded Mainnet address begins with T and is 34 bytes in length.

4.3 Structure

The three different account types are Normal, AssetIssue, and Contract. An Account contains 7 parameters:

1. **account_name**: the name for this account – e.g. BillsAccount.
2. **type**: what type of this account is – e.g. 0 (stands for type ‘Normal’).
3. **balance**: balance of this account – e.g. 4213312.

4. **vote**: received votes on this account – e.g. {("0x1b7w...9xj3",323), ("0x8djq...j12m",88),...,("0x82nd...mx6i",10001)}.
5. **asset**: other assets expected TRX in this account – e.g. {<"WishToken", 66666>, <"Dogie", 233>}.
6. **latest_operation_time**: the latest operation time of this account.

Protobuf data structure:

```
message Account {  
    message Vote {  
        bytes vote_address = 1;  
        int64 vote_count = 2;  
    }  
    bytes account_name = 1;  
    AccountType type = 2;  
    bytes address = 3;  
    int64 balance = 4;  
    repeated Vote votes = 5;  
    map<string, int64> asset = 6;  
    int64 latest_operation_time = 10;  
}
```

```
enum AccountType {  
    Normal = 0;  
    AssetIssue = 1;  
    Contract = 2;  
}
```

5. Block

A block typically contains a block header and several transactions.

Protobuf data structure:

```
message Block {  
    BlockHeader block_header = 1;  
    repeated Transaction transactions = 2;  
}
```

5.1 Block Header

A block header contains **raw_data**, **witness_signature**, and **blockID**.

Protobuf data structure:

```
message BlockHeader {  
    message raw {  
        int64 timestamp = 1;  
        bytes txTrieRoot = 2;  
        bytes parentHash = 3;  
        uint64 number = 4;  
        uint64 version = 5;  
        bytes witness_address = 6;  
    }  
    bytes witness_signature = 2;  
    bytes blockID = 3;  
}
```

5.1.1 Raw Data

Raw data is denoted as **raw_data** in Protobuf. It contains the raw data of a message, containing 6 parameters:

1. **timestamp**: timestamp of this message – e.g. 1543884429000.
2. **txTrieRoot**: the Merkle Tree's Root – e.g. 7dacsa...3ed.
3. **parentHash**: the hash of the last block – e.g. 7dacsa...3ed.
4. **number**: the block height – e.g. 4638708.
5. **version**: reserved – e.g. 5.

6. **witness_address**: the address of the witness packed in this block – e.g. 41928c...4d21.

5.1.2 Witness Signature

Witness signature is denoted as **witness_signature** in Protobuf, which is the signature for this block header from the witness node.

5.1.3 Block ID

Block ID is denoted as **blockID** in Protobuf. It contains the atomic identification of a block. A Block ID contains 2 parameters:

1. **hash**: the hash of block.
2. **number**: the hash and height of the block.

5.2 Transaction

5.2.1 Signing

TRON's transaction signing process follows a standard ECDSA cryptographic algorithm, with a SECP256K1 selection curve. A private key is a random number, and the public key is a point on the elliptic curve. The public key generation process consists of first generating a random number as a private key, and then multiplying the base point of the elliptic curve by the private key to obtain the public key. When a transaction occurs, the transaction raw data is first converted into byte format. The raw data then undergoes SHA-256 hashing. The private key corresponding to the contract address then signs the result of the SHA256 hash. The signature result is then added to the transaction.

5.2.2 Bandwidth Model

Ordinary transactions only consume bandwidth points, but smart contract operations consume both energy and bandwidth points. There are two types of bandwidth points available. Users can gain bandwidth points from freezing TRX, while 5000 free bandwidth points are also available daily.

When a TRX transaction is broadcast, it is transmitted and stored in the form of a byte array over the network. Bandwidth Points consumed by one transaction = number of transaction bytes multiplied by bandwidth points rate. For example, if the byte array length of a transaction is 200, then the transaction consumes 200 bandwidth points. However, if a TRX or token transfer results in the target account being created, then only the bandwidth points consumed to create the account will be deducted, and additional bandwidth points will not be deducted. In an account creation scenario, the network will first consume the bandwidth points that the transaction initiator gained

from freezing TRX. If this amount is insufficient, then the network consumes the transaction initiator's TRX.

In standard TRX transfer scenarios from one TRX account to another, the network first consumes the bandwidth points gained by the transaction initiator for freezing TRX. If that is insufficient, it then consumes from the free 5000 daily bandwidth points. If that is still not enough, then the network consumes the TRX of the transaction initiator. The amount is calculated by the number of bytes in the transaction multiplied by 10 SUN. Thus, for most TRX holders who may not necessarily freeze their TRX to participate in SR voting, the first step is automatically skipped (since TRX balance frozen = 0) and the 5000 daily free bandwidth powers the transaction.

For TRC-10 token transfers, the network first verifies whether the total free bandwidth points of the issued token asset are sufficient. If not, the bandwidth points obtained from freezing TRX are consumed. If there is still not enough bandwidth points, then it consumes the TRX of the transaction initiator.

5.2.3 Fee

TRON network generally does not charge fees for most transactions, however, due to system restrictions and fairness, bandwidth usage and transactions do take in certain fees.

Fee charges are broken down into the following categories:

1. Normal transactions cost bandwidth points. Users can use the free daily bandwidth points (5000) or freeze TRX to obtain more. When bandwidth points are not enough, TRX will be used directly from the sending account. The TRX needed is the number of bytes * 10 SUN.
2. Smart contracts cost energy (Section 6) but will also need bandwidth points for the transaction to be broadcasted and confirmed. The bandwidth cost is the same as above.
3. All query transactions are free. It doesn't cost energy or bandwidth.

TRON network also defines a set of fixed fees for the following transactions:

1. Creating a witness node: 9999 TRX
2. Issuing a TRC-10 token: 1024 TRX
3. Creating a new account: 0.1 TRX
4. Creating an exchange pair: 1024 TRX

5.2.4 Transaction as Proof of Stake (TaPoS)

TRON uses TaPoS to ensure the transactions all confirm the main blockchain, while making it difficult to forge counterfeit chains. In TaPoS, the networks require each transaction include part of the hash of a recent block header. This requirement prevents transactions from being replayed on forks not including the referenced block, and also signals the network that a particular user and their

stake are on a specific fork. This consensus mechanism protects the network against Denial of Service, 51%, selfish mining, and double spend attacks.

5.2.5 Transaction Confirmation

A transaction is included in a future block after being broadcast to the network. After 19 blocks are mined on TRON (including its own block), the transaction is confirmed. Each block is produced by one of the top 27 Super Representatives in a round robin fashion. Each block takes ~3 seconds to be mined on the blockchain. Time may slightly vary for each Super Representative due to network conditions and machine configurations. In general, a transaction is considered fully confirmed after ~1 minute.

5.2.6 Structure

Transaction APIs consist of the following functions:

```
message Transaction {
    message Contract {
        enum ContractType {
            AccountCreateContract = 0; // Create account/wallet
            TransferContract = 1; // Transfer TRX
            TransferAssetContract = 2; // Transfer TRC10 token
            VoteWitnessContract = 4; // Vote for Super Representative (SR)
            WitnessCreateContract = 5; // Create a new SR account
            AssetIssueContract = 6; // Create a new TRC10 token
            WitnessUpdateContract = 8; // Update SR information
            ParticipateAssetIssueContract = 9; // Purchase TRC10 token
            AccountUpdateContract = 10; // Update account/wallet information
            FreezeBalanceContract = 11; // Freeze TRX for bandwidth or energy
            UnfreezeBalanceContract = 12; // Unfreeze TRX
            WithdrawBalanceContract = 13; // Withdraw SR rewards, once per day
            UnfreezeAssetContract = 14; // Unfreeze TRC10 token
            UpdateAssetContract = 15; // Update a TRC10 token's information
            ProposalCreateContract = 16; // Create a new network proposal by any SR
            ProposalApproveContract = 17; // SR votes yes for a network proposal
            ProposalDeleteContract = 18; // Delete a network proposal by owner
            CreateSmartContract = 30; // Deploy a new smart contract
            TriggerSmartContract = 31; // Call a function on a smart contract
            GetContract = 32; // Get an existing smart contract
            UpdateSettingContract = 33; // Update a smart contract's parameters
            ExchangeCreateContract = 41; // Create a token trading pair on DEX
            ExchangeInjectContract = 42; // Inject funding into a trading pair
        }
    }
}
```

```
ExchangeWithdrawContract = 43; // Withdraw funding from a trading pair
ExchangeTransactionContract = 44; // Perform token trading
UpdateEnergyLimitContract = 45; // Update origin_energy_limit on a
smart contract
}
}
}
```

6. TRON Virtual Machine (TVM)

6.1 Introduction

TRON Virtual Machine (TVM) is a lightweight, Turing complete virtual machine developed for the TRON's ecosystem. Its goal is to provide a custom-built blockchain system that is efficient, convenient, stable, secure and scalable.

TVM initially forked from EVM¹¹ and can connect seamlessly with the existing solidity smart contract development ecosystem. Based on that, TVM additionally supports DPoS consensus.

TVM employs the concept of Energy. Different from the Gas mechanism on EVM, operations of transactions and smart contracts on TVM are free, with no TRX consumed. Technically, executable computation capacity on TVM is not restricted by total holding amount of tokens.

6.2 Workflow

The compiler first translates the Solidity smart contract into bytecode readable and executable on the TVM. The TVM then processes data through opcode, which is equivalent to operating the logic of a stack-based finite state machine. Finally, the TVM accesses blockchain data and invokes External Data Interface through the Interoperation layer.

¹¹ EVM: Ethereum Virtual Machine (<https://github.com/ethereum/ethereumj>)

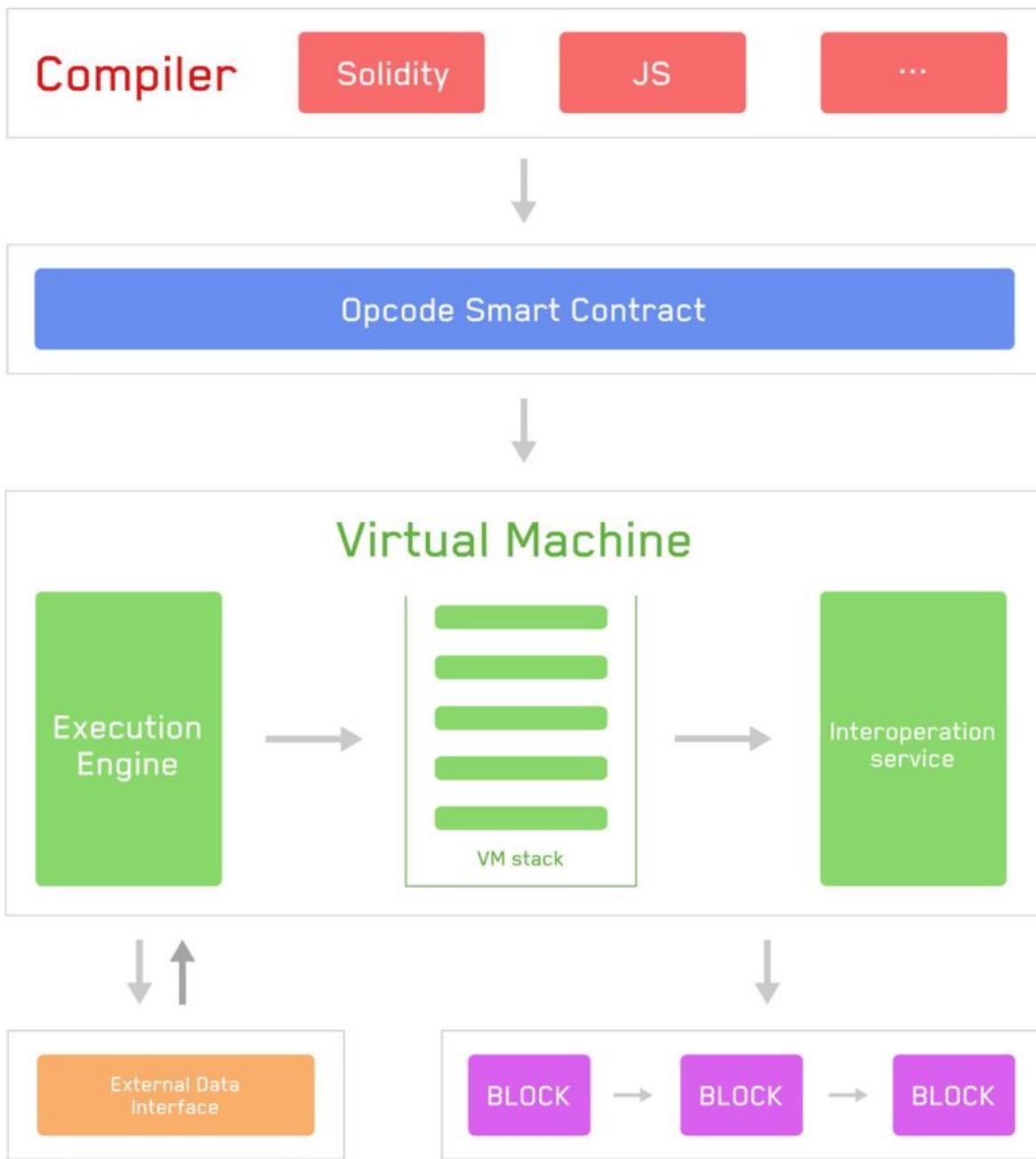


Figure 3: TVM Workflow

6.3 Performance

6.3.1 Lightweight Architecture

TVM adopts a lightweight architecture with the aim of reducing resource consumption to guarantee system performance.

6.3.2 Robust

TRX transfers and smart contract execution cost bandwidth points only, instead of TRX, which exempts TRON from being attacked. Bandwidth consumption is predictable and static since each computational step cost is fixed.

6.3.3 High Compatibility

TVM is compatible with EVM and will be compatible with more mainstream VMs in the future. Thereby, all smart contracts on EVM are executable on TVM.

6.3.4 Low Cost

Due to TVM's bandwidth setup, development costs are reduced and developers can focus on the logic development of their contract code. TVM also offers all-in-one interfaces for contract deployment, triggering and viewing to offer the convenience for developers.

7. Smart Contract

7.1 Introduction

A smart contract is a protocol that digitally verifies contract negotiation. They define the rules and penalties related to an agreement and also automatically enforce those obligations. The smart contract code facilitates, verifies, and enforces the negotiation or performance of an agreement or transaction. From a tokenization perspective, smart contracts also facilitate automatic funds transfers between participating parties should certain criteria be met.

TRON smart contracts are written in the Solidity language. Once written and tested, they can be compiled into bytecode, then deployed onto the TRON network for the TRON Virtual Machine. Once deployed, smart contracts can be queried via their contract addresses. The contract Application Binary Interface (ABI) shows the contract's call functions and is used for interacting with the network.

7.2 Energy Model

The maximum energy limit for deploying and triggering a smart contract is a function of several variables:

- Dynamic energy from freezing 1 TRX is $50,000,000,000 \text{ (Total Energy Limit)} / \text{ (Total Energy Weight)}$
- Energy limit is the daily account energy limit from freezing TRX
- Remaining daily account energy from freezing TRX is calculated as Energy Limit - Energy Used
- Fee limit in TRX is set in smart contract deploy/trigger call
- Remaining usable TRX in the account
- Energy per TRX if purchased directly ($10 \text{ SUN} = 1 \text{ Energy}$) = 100,000, SRs can vote on adjustment

There are two consumption scenarios to calculate for maximum energy limit for deployment and trigger. The logic can be expressed as follows:

```
const R = Dynamic Energy Limit
const F = Daily account energy from freezing TRX
const E = Remaining daily account energy from freezing TRX
const L = Fee limit in TRX set in deploy/trigger call
const T = Remaining usable TRX in account
```

```

const C = Energy per TRX if purchased directly

// Calculate M, defined as maximum energy limit for deployment/trigger of
smart contract
if F > L*R
    let M = min(E+T*C, L*R)
else
    let M = E+T*C

```

7.3 Deployment

When a TRON solidity smart contract is compiled, the TRON Virtual Machine reads the compiled bytecode. The bytecode consists of a section for code deployment, contract code, and the Auxdata. The Auxdata is the source code's cryptographic fingerprint, used for verification. The deployment bytecode runs the constructor function and sets up the initial storage variables. The deployment code also calculates the contract code and returns it to the TVM. The ABI is a JSON file that describes a TRON smart contract's functions. This file defines the function names, their payability, the function return values, and their state mutability.

7.4 Trigger Function

Once the TRON smart contracts are deployed, their functions can be triggered individually either via TronStudio or through API calls. State-changing functions require Energy while read-only functions execute without Energy.

7.5 TRON Solidity

TRON Solidity is a fork from Ethereum's Solidity language. TRON modifies the original project to support TRX and SUN units (1 TRX = 1,000,000 SUN). The rest of the language syntax is compatible with Solidity ^0.4.24. Thus the Tron Virtual Machine (TVM) is almost 100% compatible with EVM instructions.

8. Token

8.1 TRC-10 Token

In the TRON network, each account can issue tokens at the expense of 1024 TRX. To issue tokens, the issuer needs to specify a token name, the total capitalization, the exchange rate to TRX, circulation duration, description, website, maximum bandwidth consumption per account, total bandwidth consumption, and the amount of token frozen. Each token issuance can also configure each account's maximum daily token transfer Bandwidth Points, the entire network's maximum daily token transfer Bandwidth Points, total token supply, locking duration in days, and the total amount of tokens locked.

8.2 TRC-20 Token

TRC-20 is a technical standard used for smart contracts implementing tokens supported by the TRON Virtual Machine. It is fully compatible with ERC-20.

The interface is as follows:

```
contract TRC20Interface {  
    function totalSupply() public constant returns (uint);  
    function balanceOf(address tokenOwner) public constant returns (uint  
balance);  
    function allowance(address tokenOwner, address spender) public constant  
returns (uint remaining);  
    function transfer(address to, uint tokens) public returns (bool success);  
    function approve(address spender, uint tokens) public returns (bool  
success);  
    function transferFrom(address from, address to, uint tokens) public  
returns (bool success);  
  
    event Transfer(address indexed from, address indexed to, uint tokens);  
    event Approval(address indexed tokenOwner, address indexed spender, uint  
tokens);  
}
```

From a developer's perspective, there are several differences between TRC-10 and TRC-20. Some of the key differences are that TRC-10 tokens are accessible by APIs and smart contracts while TRC-20 tokens allow for interface customization but are only accessible within smart contracts.

From a cost perspective, TRC-10 tokens have transaction fees that are 1000 times lower than TRC-20, but carry bandwidth costs for API transfers and deposits. Transfers and deposits in smart contracts for TRC-10 tokens cost both bandwidth and energy.

8.3 Beyond

Since TRON uses the same Solidity version as Ethereum, more token standards could be readily ported to TRON.

9. Governance

9.1 Super Representative

9.1.1 General

Every account in the TRON network can apply and have the opportunity to become a Super Representative (denoted as SR). Everyone can vote for SR candidates. The top 27 candidates with the most votes will become SRs with the right and obligation to generate blocks. The votes are counted every 6 hours and the SRs will change accordingly.

To prevent malicious attacks, there is a cost to becoming an SR candidate. When applying, 9999 TRX will be burned from the applicant's account. Once successful, such account can join the SR election.

9.1.2 Election

TRON Power (denoted as TP) is needed to vote and the amount of TP depends on the voter's frozen assets (TRX).

TP is calculated in the following way:

$$1 \text{ TP} = 1 \text{ TRX frozen to get bandwidth}$$

Every account in the TRON network has the right to vote for their own SRs.

After the release (unfreeze, available after 3 days), users won't have any frozen assets and lose all TP accordingly. As a result, all votes become invalid for the ongoing and future voting round unless TRX is frozen again to vote.

Note that the TRON network only records the most recent vote, which means that every new vote will negate all previous votes.

9.1.3 Reward

a. Vote Reward

Also known as Candidate Reward, which the top 127 candidates updated once every round (6 hours) will share 115,200 TRX as mined. The reward will be split in accordance with the vote weight each candidate receives. Each year, the total reward for candidates will be 168,192,000 TRX.

Total vote reward per round

Why 115,200 TRX every round?

$$115,200 \text{ TRX} = \text{total vote reward per round (VR/round)}$$

$$VR/\text{round} = 16 \text{ TRX/block} \times 20 \text{ blocks/min} \times 60 \text{ mins/hr} \times 6 \text{ hrs/round}$$

Notice: this is set by WITNESS_STANDBY_ALLOWANCE = 115,200 TRX. See dynamic network parameters.

Total vote reward per year

Why 168,192,000 TRX every year?

$$168,192,000 \text{ TRX} = \text{total vote reward per year (VR/year)}$$

$$VR/\text{year} = 115,200 \text{ TRX/round} \times 4 \text{ rounds/day} \times 365 \text{ days/year}$$

b. Block Reward

Also known as Super Representative Reward, which the top 27 candidates (SRs) who are elected every round (6 hours) will share roughly 230,400 TRX as mined. The reward will be split evenly between the 27 SRs (minus the total reward blocks missed due to network error). A total of 336,384,000 TRX will be awarded annually to the 27 SRs.

Total block reward per round

Why 230,400 TRX every round?

$$230,400 \text{ TRX} = \text{total block reward per round (BR/round)}$$

$$BR/\text{round} = 32 \text{ TRX/bloc} \times 20 \text{ blocks/min} \times 60 \text{ mins/hr} \times 6 \text{ hrs/round}$$

Notice: the unit block reward is set by WITNESS_PAY_PER_BLOCK = 32 TRX. See dynamic network parameters.

Total block reward per year

Why 336,384,000 TRX every year?

$$336,384,000 \text{ TRX} = \text{total block reward per year (BR/year)}$$

$$BR/\text{year} = 230,400 \text{ TRX/round} \times 4 \text{ rounds/day} \times 365 \text{ days/year}$$

January 1, 2021

There will be no inflation on the TRON network before January 1, 2021, and the TRON Foundation will award all block rewards and candidate rewards prior to that date.

c. Reward Calculation

SR reward calculation

total reward = vote reward (VR) + block reward (BR)

VR = total VR × $\frac{\text{votes SR candidate received}}{\text{total votes}}$

BR = $\frac{\text{total BR}}{27}$ - block missed × 32

Note: the reward is calculated per SR per round (6 hours)

Rank 28 to rank 127 SR candidate reward calculation

total reward = vote reward (VR)

VR = total VR × $\frac{\text{votes SR candidate received}}{\text{total votes}}$

Note: the reward is calculated per SR candidate per round (6 hours)

9.2 Committee

9.2.1 General

The committee is used to modify TRON dynamic network parameters, such as block generation rewards, transaction fees, etc. The committee consists of the 27 SRs in the current round. Each SR has the right to propose and vote on proposals. When a proposal receives 19 votes or more, it is approved and the new network parameters will be applied in the next maintenance period (3 days).

9.2.2 Dynamic Network Parameters

0. MAINTENANCE_TIME_INTERVAL

a. Description

Modify the maintenance interval time in ms. Known as the SR vote interval time per round.

b. Example

[6 * 3600 * 1000] ms - which is 6 hours.

c. Range

[3 * 27* 1000, 24 * 3600 * 1000] ms

1. ACCOUNT_UPGRADE_COST

a. Description

Modify the cost of applying for SR account.

b. Example

[9,999,000,000] SUN - which is 9,999 TRX.

c. Range

[0,100 000 000 000 000 000] SUN

2. CREATE_ACCOUNT_FEE

a. Description

Modify the account creation fee.

- b. Example
[100,000] SUN - which is 1 TRX.
 - c. Range
[0,100 000 000 000 000 000] SUN
3. TRANSACTION_FEE
- a. Description
Modify the amount of fee used to gain extra bandwidth.
 - b. Example
[10] SUN/byte.
 - c. Range
[0,100 000 000 000 000 000] SUN/byte
4. ASSET_ISSUE_FEE
- a. Description
Modify asset issuance fee.
 - b. Example
[1024,000,000] SUN - which is 1024 TRX.
 - c. Range
[0,100 000 000 000 000 000] SUN
5. WITNESS_PAY_PER_BLOCK
- a. Description
Modify SR block generation reward. Known as unit block reward.
 - b. Example
[32,000,000] SUN - which is 32 TRX.
 - c. Range
[0,100 000 000 000 000 000] SUN
6. WITNESS_STANDBY_ALLOWANCE
- a. Description
Modify the rewards given to the top 127 SR candidates. Known as total vote reward per round.
 - b. Example
[115,200,000,000] SUN - which is 115,200 TRX.
 - c. Range
[0,100 000 000 000 000 000] SUN
7. CREATE_NEW_ACCOUNT_FEE_IN_SYSTEM_CONTRACT
- a. Description
Modify the cost of account creation. Combine dynamic network parameters #8 to get total account creation cost:
 $CREATE_NEW_ACCOUNT_FEE_IN_SYSTEM_CONTRACT \times CREATE_NEW_ACCOUNT_BANDWIDTH_RATE$
 - b. Example
[0] SUN.
 - c. Range
[0,100 000 000 000 000 000] SUN
8. CREATE_NEW_ACCOUNT_BANDWIDTH_RATE

a. Description

Modify the cost of account creation. Combine dynamic network parameters #7 to get total account creation cost:

CREATE_NEW_ACCOUNT_FEE_IN_SYSTEM_CONTRACT × CREATE_NEW_ACCOUNT_BANDWIDTH_RATE

b. Example

[1].

c. Range

[0,100,000,000,000,000,000]

9. ALLOW_CREATION_OF_CONTRACTS

a. Description

To turn on Tron Virtual Machine (TVM).

b. Example

True - set to activate and effect since 10/10/2018 23:47 UTC.

c. Range

True/False

10. REMOVE_THE_POWER_OF_THE_GR

a. Description

Remove the initial GR genesis votes

b. Example

True - effected at 11/4/2018 08:46 UTC.

c. Range

True/False - Notice: cannot set back to False from True.

11. ENERGY_FEE

a. Description

Modify the fee of 1 energy.

b. Example

20 SUN.

c. Range

[0,100 000 000 000 000 000] SUN

12. EXCHANGE_CREATE_FEE

a. Description

Modify the cost of trading pair creation. Known as the cost of creating a trade order.

b. Example

[1,024,000,000] SUN - which is 1024 TRX.

c. Range

[0,100 000 000 000 000 000] SUN

13. MAX_CPU_TIME_OF_ONE_TX

a. Description

Modify the maximum execution time of one transaction. Known as the timeout limit of one transaction.

b. Example

50 ms.

c. Range

[0, 1000] ms

14. ALLOW_UPDATE_ACCOUNT_NAME

a. Description

Modify the option to let an account update their account name.

b. Example

False - which is available to propose from java-tron Odyssey v3.2.

c. Range

True/False - Notice: cannot set back to False from True.

15. ALLOW_SAME_TOKEN_NAME

a. Description

Modify the validation of allowing different token have a duplicate name.

b. Example

False - which is available to propose from java-tron Odyssey v3.2.

c. Range

True/False - Notice: cannot set back to False from True.

16. ALLOW_DELEGATE_RESOURCE

a. Description

Modify the validation of allowing to issue token with a duplicate name, so the **tokenId** of the token, in long integer data type, would be the only atomic identification of a token.

b. Example

False - which is available to propose from java-tron Odyssey v3.2.

c. Range

True/False - Notice: cannot set back to False from True.

17. TOTAL_ENERGY_LIMIT

a. Description

Modify the whole network total energy limit.

b. Example

[50,000,000,000,000,000] SUN - which is 50,000,000,000 TRX.

c. Range

[0,100,000,000,000,000,000] SUN

18. ALLOW_TVM_TRANSFER_TRC10

a. Description

Allow TRC-10 token transfer within smart contracts.

ALLOW_UPDATE_ACCOUNT_NAME, ALLOW_SAME_TOKEN_NAME, ALLOW_DELEGATE_RESOURCE proposals must all be approved before proposing this parameter change.

b. Example

False - which is available to propose from java-tron Odyssey v3.2.

c. Range

True/False - Notice: cannot set back to False from True.

9.2.3 Create Proposal

Only the SR accounts have the rights to propose a change in dynamic network parameters.

9.2.4 Vote Proposal

Only committee members (SRs) can vote for a proposal and the member who does not vote in time will be considered as a disagree. The proposal is active for 3 days after it is created. The vote can be changed or retrieved during the 3-days voting window. Once the period ends, the proposal will either succeed (19+ votes) or fail (and end).

9.2.5 Cancel Proposal

The proposer can cancel the proposal before it becomes effective.

9.3 Structure

SRs are the witnesses of newly generated blocks. A witness contains 8 parameters:

1. **address**: the address of this witness – e.g. 0xu82h...7237.
2. **voteCount**: number of received votes on this witness – e.g. 234234.
3. **pubKey**: the public key for this witness – e.g. 0xu82h...7237.
4. **url**: the url for this witness – e.g. <https://www.noonetrust.com>.
5. **totalProduced**: the number of blocks this witness produced – e.g. 2434.
6. **totalMissed**: the number of blocks this witness missed – e.g. 7.
7. **latestBlockNum**: the latest height of block – e.g. 4522.
8. **isjobs**: a boolean flag.

Protobuf data structure:

```
message Witness{  
    bytes address = 1;  
    int64 voteCount = 2;  
    bytes pubKey = 3;  
    string url = 4;  
    int64 totalProduced = 5;  
    int64 totalMissed = 6;  
    int64 latestBlockNum = 7;  
    bool isJobs = 8;  
}
```

10. DApp Development

10.1 APIs

The TRON network offers a wide selection of over 60+ HTTP API gateways for interacting with the network via Full and Solidity Nodes. Additionally, TronWeb is a comprehensive JavaScript library containing API functions that enable developers to deploy smart contracts, change the blockchain state, query blockchain and contract information, trade on the DEX, and much more. These API gateways can be directed towards a local privatenet, the Shasta testnet, or the TRON Mainnet.

10.2 Networks

TRON has both a Shasta testnet as well as a Mainnet. Developers may connect to the networks by deploying nodes, interacting via TronStudio, or using APIs via the TronGrid service. The TronGrid service consists of load balanced node clusters hosted on AWS servers worldwide. As DApp development scales up and API call volumes increase, TronGrid successfully fields the increase in API traffic.

10.3 Tools

TRON offers a suite of development tools for enabling developers to create innovative DApps. TronBox is a framework that allows developers to test and deploy smart contracts via the TronWeb API. TronGrid is a load balanced and hosted API service that allows developers to access the TRON network without having to run their own node. TronGrid offers access to both the Shasta testnet as well as the TRON Mainnet. TronStudio is a comprehensive Integrated Development Environment (IDE) that enables developers to compile, deploy, and debug their Solidity smart contracts. TronStudio contains an internal full node that creates a private local environment for smart contract testing prior to deployment. The TronWeb API library connects developers to the network via a wide selection of HTTP API calls wrapped in JavaScript.

10.4 Resources

The TRON Developer Hub is a comprehensive API documentation¹² site tailored towards developers wishing to build on the TRON network. The Developer Hub provides a high-level conceptual understanding of TRON and walks users through the details of interacting with the

¹² Developer Hub: <https://developers.tron.network/>

network. The guides walk developers through node setup, deployment and interaction with smart contracts, API interaction and implementation, building sample DApps, and using each of the developer tools. Additionally, developer community channels are available through Discord¹³.

¹³ Discord: <https://discordapp.com/invite/GsRgsTD>

11. Conclusion

TRON is a scalable blockchain solution that has employed innovative methods for tackling challenges faced by legacy blockchain networks. Having reached over 2M transactions per day, with over 700K TRX accounts, and surpassing 2000 TPS, TRON has enabled the community in creating a decentralized and democratized network.

Tezos — a self-amending crypto-ledger

White paper

L.M Goodman

September 2, 2014

Changes between the original paper and our current implementation are indicated in red.

“Our argument is not flatly circular, but something like it.”
— Willard van Orman Quine

Abstract

We present Tezos, a generic and self-amending crypto-ledger. Tezos can instantiate any blockchain based ledger. The operations of a regular blockchain are implemented as a purely functional module abstracted into a shell responsible for network operations. Bitcoin, Ethereum, Cryptonote, etc. can all be represented within Tezos by implementing the proper interface to the network layer.

Most importantly, Tezos supports meta upgrades: the protocols can evolve by amending their own code. To achieve this, Tezos begins with a seed protocol defining a procedure for stakeholders to approve amendments to the protocol, *including* amendments to the voting procedure itself. This is not unlike philosopher Peter Suber’s Nomic[3], a game built around a fully introspective set of rules.

In addition, Tezos’s seed protocol is based on a pure proof-of-stake system and supports Turing complete smart contracts. Tezos is implemented in OCaml, a powerful functional programming language offering speed, an unambiguous syntax and semantic, and an ecosystem making Tezos a good candidate for formal proofs of correctness.

Familiarity with the Bitcoin protocol and basic cryptographic primitives are assumed in the rest of this paper.

Contents

1	Introduction	3
2	Self-amending cryptoledger	3
2.1	Mathematical representation	3
2.2	The network shell	4
2.2.1	Clock	4
2.2.2	Chain selection algorithm	4
2.2.3	Network level defense	5
2.3	Functional representation	5
2.3.1	Validating the chain	5
2.3.2	Amending the protocol	6
2.3.3	RPC	7
3	Seed protocol	8
3.1	Economy	8
3.1.1	Coins	8
3.1.2	Mining and signing rewards	8
3.1.3	Lost coins	9
3.1.4	Amendment rules	9
3.2	Proof-of-stake mechanism	10
3.2.1	Overview	10
3.2.2	Clock	11
3.2.3	Generating the random seed	11
3.2.4	Follow-the-coin procedure	12
3.2.5	Mining blocks	13
3.2.6	Signing blocks	13
3.2.7	Weight of the chain	14
3.2.8	Denunciations	14
3.3	Smart contracts	14
3.3.1	Contract type	14
3.3.2	Origination	15
3.3.3	Transactions	15
3.3.4	Storage fees	16
3.3.5	Code	16
3.3.6	Fees	16
4	Conclusion	17

1 Introduction

In the first part of this paper, we will discuss the concept of abstract blockchains and the implementation of a self-amending crypto-ledger. In the second part, we will describe our proposed seed protocol.

2 Self-amending cryptoledger

A blockchain protocol can be decomposed into three distinct protocols:

- The network protocol discovers blocks and broadcasts transactions.
- The transaction protocol specifies what makes a transaction valid.
- The consensus protocol forms consensus around a unique chain.

Tezos implements a generic network shell. This shell is agnostic to the transaction protocol and to the consensus protocol. We refer to the transaction protocol and the consensus protocol together as a “blockchain protocol”. We will first give a mathematical representation of a blockchain protocol and then describe some of the implementation choices in Tezos.

2.1 Mathematical representation

A blockchain protocol is fundamentally a monadic implementation of concurrent mutations of a global state. This is achieved by defining “blocks” as operators acting on this global state. The free monoid of blocks acting on the genesis state forms a tree structure. A global, canonical, state is defined as the minimal leaf for a specified ordering.

This suggests the following abstract representation:

- Let (\mathbf{S}, \leq) be a totally ordered, countable, set of possible states.
- Let $\emptyset \notin \mathbf{S}$ represent a special, invalid, state.
- Let $\mathbf{B} \subset \mathbf{S}^{\mathbf{S} \cup \{\emptyset\}}$ be the set of blocks. The set of *valid* blocks is $\mathbf{B} \cap \mathbf{S}^{\mathbf{S}}$.

The total order on \mathbf{S} is extended so that $\forall s \in \mathbf{S}, \emptyset < s$. This order determines which leaf in the block tree is considered to be the canonical one. Blocks in \mathbf{B} are seen as operators acting on the state.

All in all, any blockchain protocol¹ (be it Bitcoin, Litecoin, Peercoin, Ethereum, Cryptonote, etc) can be fully determined by the tuple:

$$(\mathbf{S}, \leq, \emptyset, \mathbf{B} \subset \mathbf{S}^{\mathbf{S} \cup \{\emptyset\}})$$

¹GHOST is an approach which orders the leafs based on properties of the tree. Such an approach is problematic for both theoretical and practical reasons. It is almost always better to emulate it by inserting proofs of mining in the main chain.

The networking protocol is fundamentally identical for these blockchains. “Mining” algorithms are but an emergent property of the network, given the incentives for block creation.

In Tezos, we make a blockchain protocol introspective by letting blocks act on the protocol itself. We can then express the set of protocols recursively as

$$\mathcal{P} = \left\{ \left(\mathbf{S}, \leq, \emptyset, \mathbf{B} \subset \mathbf{S}^{(\mathbf{S} \times \mathcal{P}) \cup \{\emptyset\}} \right) \right\}$$

2.2 The network shell

This formal mathematical description doesn’t tell us *how* to build the block tree. This is the role of the network shell, which acts as an interface between a gossip network and the protocol.

The network shell works by maintaining the best chain known to the client. It is aware of three type of objects. The first two are transactions and blocks, which are only propagated through the network if deemed valid. The third are protocols, OCaml modules used to amend the existing protocol. They will be described in more details later on. For now we will focus on transaction and blocks.

The most arduous part of the network shell is to protect nodes against denial-of-service attacks.

2.2.1 Clock

Every block carries a timestamp visible to the network shell. Blocks that appear to come from the future are buffered if their timestamps are within a few minutes of the system time and rejected otherwise. The protocol design must tolerate reasonable clock drifts in the clients and must assume that timestamps can be falsified.

2.2.2 Chain selection algorithm

The shell maintains a single chain rather than a full tree of blocks. This chain is only overwritten if the client becomes aware of a strictly better chain.

Maintaining a tree would be more parsimonious in terms of network communications but would be susceptible to denial-of-service attacks where an attacker produces a large number of low-scoring but valid forks.

Yet, it remains possible for a node to lie about the score of a given chain, a lie that the client may only uncover after having processed a potentially large number of blocks. However, such a node can be subsequently ignored.

Fortunately, a protocol can have the property that low scoring chains exhibit a low rate of block creation. Thus, the client would only consider a few blocks of a “weak” fork before concluding that the announced score was a lie.

2.2.3 Network level defense

In addition, the shell is “defensive”. It attempts to connect to many peers across various IP ranges. It detects disconnected peers and bans malicious nodes.

To protect against certain denial of service attacks, the protocol provides the shell with context dependent bounds on the size of blocks and transactions.

2.3 Functional representation

2.3.1 Validating the chain

We can efficiently capture almost all the genericity of our abstract blockchain structure with the following OCaml types. To begin with, a block header is defined as:

```
type raw_block_header = {
  pred: Block_hash.t;
  header: Bytes.t;
  operations: Operation_hash.t list;
  timestamp: float;
}
```

We are purposefully not typing the header field more strongly so it can represent arbitrary content. However, we do type the fields necessary for the operation of the shell. These include the hash of the preceding block, a list of operation hashes and a timestamp. In practice, the operations included in a block are transmitted along with the blocks at the network level. Operations themselves are represented as arbitrary blobs.

```
type raw_operation = Bytes.t
```

The state is represented with the help of a **Context** module which encapsulates a disk-based immutable key-value store. The structure of a key-value store is versatile and allows us to efficiently represent a wide variety of states.

```
module Context = sig
  type t
  type key = string list

  val get: t -> key -> Bytes.t option Lwt.t
  val set: t -> key -> Bytes.t -> t Lwt.t
  val del: t -> key -> t Lwt.t
  (*...*)
end
```

To avoid blocking on disk operations, the functions use the asynchronous monad Lwt[4]. Note that the operations on the context are purely functional: **get** uses the **option** monad rather than throwing an exception while **set** and **del** both return a new **Context**. The **Context** module uses a combination of memory caching and disk storage to efficiently provide the appearance of an immutable store.

We can now define the module type of an arbitrary blockchain protocol:

```

type score = Bytes.t list
module type PROTOCOL = sig
  type operation
  val parse_block_header : raw_block_header -> block_header option
  val parse_operation : Bytes.t -> operation option

  val apply :
    Context.t ->
    block_header option ->
    (Operation_hash.t * operation) list ->
    Context.t option Lwt.t

  val score : Context.t -> score Lwt.t
  (*...*)
end

```

We no longer compare states directly as in the mathematical model, instead we project the **Context** onto a list of bytes using the **score** function. List of bytes are ordered first by length, then by lexicographic order. This is a fairly generic structure, similar to the one used in software versioning, which is quite versatile in representing various orderings.

Why not define a comparison function within the protocol modules? First off it would be hard to enforce the requirement that such a function represent a *total* order. The score projection always verifies this (ties can be broken based on the hash of the last block). Second, in principle we need the ability to compare states across distinct protocols. Specific protocol amendment rules are likely to make this extremely unlikely to ever happen, but the network shell does not know that.

The operations **parse_block_header** and **parse_operation** are exposed to the shell and allow it to pass fully typed operations and blocks to the protocol but also to check whether these operations and blocks are well-formed, before deciding to relay operations or to add blocks to the local block tree database.

The apply function is the heart of the protocol:

- When it is passed a block header and the associated list of operations, it computes the changes made to the context and returns a modified copy. Internally, only the difference is stored, as in a versioning system, using the block's hash as a version handle.
- When it is only passed a list of operations, it greedily attempts to apply as many operations as possible. This function is not necessary for the protocol itself but is of great use to miners attempting to form valid blocks.

2.3.2 Amending the protocol

Tezos's most powerful feature is its ability to implement protocol capable of self-amendment. This is achieved by exposing two procedures functions to the protocol:

- **set_test_protocol** which replaces the protocol used in the testnet with

a new protocol (typically one that has been adopted through a stakeholder voter).

- **promote_test_protocol** which replaces the current protocol with the protocol currently being tested

These functions transform a Context by changing the associated protocol. The new protocol takes effect when the following block is applied to the chain.

```
module Context = sig
  type t
  (*...*)
  val set_test_protocol: t -> Protocol_hash.t Lwt.t
  val promote_test_protocol: t -> Protocol_hash.t -> t Lwt.t
end
```

The **protocol_hash** is the **sha256** hash of a tarball of **.ml** and **.mli** files. These files are compiled on the fly. They have access to a small standard library but are sandboxed and may not make any system call.

These functions are called through the **apply** function of the protocol which returns the new **Context**.

Many conditions can trigger a change of protocol. In its simplest version, a stakeholder vote triggers a change of protocol. More complicated rules can be progressively voted in. For instance, if the stakeholder desire they may pass an amendment that will require further amendments to provide a computer checkable proof that the new amendment respects certain properties. This is effectively and algorithmic check of “constitutionality”.

2.3.3 RPC

In order to make the GUI building job’s easier, the protocol exposes a JSON-RPC API. The API itself is described by a json schema indicating the types of the various procedures. Typically, functions such as **get_balance** can be implemented in the RPC.

```
type service = {
  name : string list ;
  input : json_schema option ;
  output : json_schema option ;
  implementation : Context.t -> json -> json option Lwt.t
}
```

The name is a list of string to allow namespaces in the procedures. Input and output are optionally described by a json schema.

Note that the call is made on a given context which is typically a recent ancestor of the highest scoring leaf. For instance, querying the context six blocks above the highest scoring leaf displays the state of the ledger with six confirmations.

The UI itself can be tailored to a specific version of the protocol, or generically derived from the JSON specification.

3 Seed protocol

Much like blockchains start from a genesis hash, Tezos starts with a seed protocol. This protocol can be amended to reflect virtually any blockchain based algorithm.

3.1 Economy

3.1.1 Coins

There are initially 10 000 000 000 (ten billion) coins ([the initial extent of the token supply will be the number of tokens issued during the crowdsale and not specifically “10 billion”, which was merely a placeholder. This change in size has no effect on the principal at hand](#)), divisible up to two decimal places (for the sake of precision we may in actuality be using eight digits after the decimal). We suggest that a single coin be referred to as a “tez” and that the smallest unit simply as a cent. We also suggest to use the symbol tz (\u20ac729, “Latin small letter tz”) to represent a tez. Therefore 1 cent = $\text{tz}0.01$ = one hundredth of a tez.

3.1.2 Mining and signing rewards

Principle We conjecture that the security of any decentralised currency requires to incentivize the participants with a pecuniary reward ([we are in the process of finalizing the rewards schedule at the moment](#)). As explained in the position paper, relying on transaction costs alone suffers from a tragedy of the commons. In Tezos, we rely on the combination of a bond and a reward.

Bonds are one year ([bonds will now only last a single cycle, given the high opportunity cost and little benefit to security of extending the bonding period past one cycle](#)) security deposits purchased by miners ([endorsers will also be required to purchase bonds](#)). In the event of a double signing, these bonds are forfeited.

After a year ([cycle](#)), the miners ([and endorsers](#)) receive a reward along with their bond to compensate for their opportunity cost. The security is primarily being provided by the value of the bond and the reward need only be a small percentage of that value.

The purpose of the bonds is to diminish the amount of reward needed, and perhaps to use the loss aversion effect to the network’s advantage.

Specifics In the seed protocol, mining a block offers a reward of $\text{tz}512$ and requires a bond of $\text{tz}1536$. Signing a block offers a reward of $32\Delta T^{-1}$ tez where ΔT is the time interval in minutes between the block being signed and its predecessor. There are up to 16 signatures per block and signing requires no bond. [These numbers were based on a supply of 10 billions tokens and we will tweak them accordingly. We may increase the number of signatures per block as well as we’ve found in simulations it can strongly increase the difficulty of forks.](#)

Thus, assuming a mining rate of one block per minute, about 8% of the initial money mass should be held in the form of safety bonds after the first year. **subject to change based on the adjustment of the parameters above.**

The reward schedule implies at most a 5.4% *nominal* inflation rate (the total block rewards will still start at about 5% per year, but we may add an asymptotic cap to the total number of tokens. We think it's irrelevant when the governance model is aligned with the token holder's interest, but it is important to some people so we're reluctantly considering it). *Nominal* inflation is neutral, it neither enriches nor impoverishes anyone².

Note that the period of a year is determined from the block's timestamps, not from the number of blocks. This is to remove uncertainty as to the length of the commitment made by miners.

Looking forward The proposed reward gives miners a 33% return on their bond (we're currently revising these parameters but will soon finalize a method that makes sense for all parties). This return needs to be high in the early days as miners and signers commit to hold a potentially volatile asset for an entire year (bonds will only last for a cycle and not a full year).

However, as Tezos mature, this return could be gradually lowered to the prevailing interest rate. A nominal rate of inflation below 1% could safely be achieved, though it's not clear there would be any point in doing so.

3.1.3 Lost coins

In order to reduce uncertainty regarding the monetary mass, addresses showing no activity for over a year (as determined by timestamps) are destroyed along with the coins they contain (inactive addresses will no longer lose their funds after one year as initially proposed in the white paper, they will only lose their staking rights until they become active again. What it means is that if an address is inactive, it will not be selected to create blocks (which would slow down the consensus algorithm), and it will not be allowed to vote until it is reactivated (to avoid uncertainty about participation rate)).

3.1.4 Amendment rules

Amendments are adopted over election cycles lasting $N = 2^{17} = 131\,072$ blocks each. Given the a one minute block interval, this is about three calendar months. The election cycle is itself divided in four quarters of $2^{15} = 32\,768$ blocks. This cycle is relatively short to encourage early improvements, but it is expected that further amendments will increase the length of the cycle (protocol upgrade votes will be much more frequent in the first year in order to allow for rapid iteration. As a security measure, the Tezos foundation will have a veto power expiring after twelve months, until we rule out any kinks in the voting procedure). Adoption requires a certain quorum to be met. This quorum starts at $Q = 80\%$ but

²In contrast, Bitcoin's mining inflation impoverishes Bitcoin holders as a whole, and central banking enriches the financial sector at the expense of savers

dynamically adapts to reflect the average participation. This is necessary if only to deal with lost coins.

First quarter Protocol amendments are suggested by submitting the hash of a tarball of `.ml` and `.mli` files representing a new protocol. Stakeholders may approve of any number of these protocols. This is known as “approval voting”, a particularly robust voting procedure.

Second quarter The amendment receiving the most approval in the first quarter is now subject to a vote. Stakeholders may cast a vote for, against or can choose to explicitly abstain. Abstentions count towards the quorum.

Third quarter If the quorum is met (including explicit abstentions), and the amendment received 80% of yays, the amendment is approved and replaces the test protocol. Otherwise, it is rejected. Assuming the quorum reached was q , the minimum quorum Q is updated as such:

$$Q \leftarrow 0.8Q + 0.2q.$$

The goal of this update is to avoid lost coins causing the voting procedure to become stuck over time. The minimum quorum is an exponential moving average of the quorum reached over each previous election.

Fourth quarter Assuming the amendment was approved, it will have been running in the testnet since the beginning of the third quarter. The stakeholders vote a second time to confirm they wish to promote the test protocol to the main protocol. This also requires the quorum to be met and an 80% supermajority.

We deliberately chose a conservative approach to amendments. However, stakeholders are free to adopt amendments loosening or tightening this policy should they deem it beneficial

3.2 Proof-of-stake mechanism

3.2.1 Overview

Our proof-of-stake mechanism is a mix of several ideas, including Slasher[1], chain-of-activity[2], and proof-of-burn. The following is a brief overview of the algorithm, the components of which are explained in more details below.

Each block is mined by a random stakeholder (the miner) and includes multiple signatures of the previous block provided by random stakeholders (the signers). Mining and signing both offer a small reward but also require making a one year (**unbonding will happen after one cycle, and not one year as initially suggested. Prolonging the period longer than a cycle did not really improve security at the cost of immobilizing a lot of capital**) safety deposit to be forfeited in the event of a double mining or double signing.

The protocol unfolds in cycles of 2048 blocks. At the beginning of each cycle, a random seed is derived from numbers that block miners chose and committed to in the penultimate cycle, and revealed in the last. Using this random seed, a follow the coin strategy is used to allocate mining rights and signing rights to a specific addresses for the next cycle. See figure 1.

3.2.2 Clock

The protocol imposes minimum delays between blocks. In principle, each block can be mined by any stakeholder. However, for a given block, each stakeholder is subject to a random minimum delay. The stakeholder receiving the highest priority may mine the block one minute after the previous block. The stakeholder receiving the second highest priority may mine the block two minutes after the previous block, the third, three minutes, and so on.

This guarantees that a fork where only a small fraction of stakeholder contribute will exhibit a low rate of block creation. If this weren't the case, a CPU denial of service attacks would be possible by tricking nodes into verifying a very long chain claimed to have a very high score.

3.2.3 Generating the random seed

Every block mined carries a hash commitment to a random number chosen by the miner. These numbers must be revealed in the next cycle under penalty of forfeiting the safety bond. This harsh penalty is meant to prevent selective withholding of the numbers which could be sued to attack the entropy of the seed.

Malicious miners in the next cycle could attempt to censor such reveals, however since multiple numbers may be revealed in a single block, they are very unlikely to succeed.

All the revealed numbers in a cycle are combined in a hash list and the seed is derived from the root using the `scrypt` key derivation function. The key derivation should be tuned so that deriving the seed takes on the order of a fraction of a percent of the average validation time for a block on a typical desktop PC.

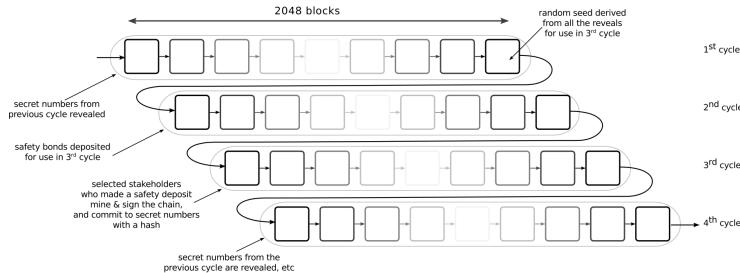


Figure 1: Four cycles of the proof-of-stake mechanism

3.2.4 Follow-the-coin procedure

In order to randomly select a stakeholder, we use a follow the coin procedure.

Principle The idea is known in bitcoin as follow-the-satoshi. The procedures works “as-if” every satoshi ever minted had a unique serial number. Satoshis are implicitly ordered by creation time, a random satoshi is drawn and tracked through the blockchain. Of course, individual cents are not tracked directly. Instead, rules are applied to describe what happens when inputs are combined and spent over multiple output.

In the end, the algorithm keeps track of a set of intervals associated with each key. Each intervals represents a “range” of satoshis. Unfortunately, over time, the database becomes more and more fragmented, increasing bloat on the client side.

Coin Rolls We optimize the previous algorithm by constructing large “coin rolls” made up of 10 000 tez. There are thus about one million rolls in existence. A database maps every roll to its current owner.

Each address holds a certain set of specific rolls as well as some loose change. When we desire to spend a fraction of a full roll, the roll is broken and its serial number is sent in a LIFO queue of rolls, a sort of “limbo”. Every transaction is processed in a way that minimizes the number of broken rolls. Whenever an address holds enough coins to form a roll, a serial number is pulled from the queue and the roll is formed again.

The LIFO priority ensures that an attacker working on a secret fork cannot change the coins he holds by shuffling change between accounts.

A slight drawback of this approach is that stake is rounded down to the nearest integer number of rolls. However, this provides a massive improvement in efficiency over the follow-the-satoshi approach.

While the rolls are numbered, this approach does not preclude the use of fungibility preserving protocols like Zerocash. Such protocols can use the same “limbo” queue technique.

Motivation This procedure is functionally different from merely drawing a random address weighted by balance.

Indeed, in a secretive fork, a miner could attempt to control the generation of the random seed and to assign itself signing and minting rights by creating the appropriate addresses ahead of time. This is much harder to achieve if rolls are randomly selected, as the secretive fork cannot fake ownership of certain rolls and must thus try to preimage the hash function applied to the seed to assign itself signing and minting rights.

Indeed, in a cycle of length $N = 2048$, someone holding a fraction f of the rolls will receive on average fN mining rights, and the effective fraction received,

f_0 will have a standard deviation of

$$\sqrt{\frac{1}{N}} \sqrt{\frac{1-f}{f}}.$$

If an attacker can perform a brute-force search through W different seeds, then his expected advantage is at most³

$$\left(\sqrt{\frac{2 \log(W)}{N}} \sqrt{\frac{1-f}{f}} \right) f N$$

blocks. For instance, an attacker controlling $f = 10\%$ of the rolls should expect to mine about 205 blocks per cycle. In a secret fork where he attempts to control the seed, assuming he computed over a trillion hashes, he could assign itself about 302 blocks, or about 14.7% of the blocks. Note that:

- The hash from which the seed is derived is an expensive key derivation function, rendering brute-force search impractical.
- To make linear gains in blocks mined, the attacked needs to expend a quadratically exponential effort.

3.2.5 Mining blocks

The random seed is used to repeatedly select a roll. The first roll selected allows its stakeholder to mine a block after one minute, the second one after two minutes — and so on.

When a stakeholder observes the seed and realizes he can mint a high priority block in the next cycle, he can make a security deposit.

To avoid a potentially problematic situation were no stakeholder made a safety deposit to mine a particular block, after a 16 minutes delay, the block may be mined without a deposit.

Bonds are implicitly returned to their buyers immediately in any chain where they do not mine the block.

3.2.6 Signing blocks

As it is, we almost have a working proof of stake system. We could define a chain's weight to be the number of blocks. However, this would open the door to a form of selfish mining.

We thus introduce a signing scheme. While a block is being minted, the random seed is used to randomly assign 16 signing rights to 16 rolls.

The stakeholders who received signing rights observe the blocks being minted and then submit signatures of that blocks. Those signatures are then included

³this is a standard bound on the expectation of the maximum of W normally distributed variable

in the next block, by miners attempting to secure their parent's inclusion in the blockchain.

The signing reward received by signers is inversely proportional to the time interval between the block and its predecessor.

Signers thus have a strong incentive to sign what they genuinely believe to be the best block produced at one point. They also have a strong incentive to agree on which block they will sign as signing rewards are only paid if the block ends up included in the blockchain.

If the highest priority block isn't mined (perhaps because the miner isn't online), there could be an incentive for signers to wait for a while, just in case the miner is late. However, other signers may then decide to sign the best priority block, and a new block could include those signatures, leaving out the holdouts. Thus, miners are unlikely to follow this strategy.

Conversely, we could imagine an equilibrium where signers panic and start signing the first block they see, for fear that other signers will do so and that a new block will be built immediately. This is however a very contrived situation which benefits no one. There is no incentive for signers to think this equilibrium is likely, let alone to modify the code of their program to act this way. A malicious stakeholder attempting to disrupt the operations would only hurt itself by attempting to follow this strategy, as others would be unlikely to follow suit.

3.2.7 Weight of the chain

The weight is the number of signatures.

3.2.8 Denunciations

In order to avoid the double minting of a block or the double signing of a block, a miner may include in his block a denunciation.

This denunciation takes the form of two signatures. Each minting signature or block signature signs the height of the block, making the proof of malfeasance quite concise.

While we could allow anyone to denounce malfeasance, there is really no point to allow anyone else beyond the block miner. Indeed, a miner can simply copy any proof of malfeasance and pass it off as its own discovery.⁴

Once a party has been found guilty of double minting or double signing, the safety bond is forfeited.

3.3 Smart contracts

3.3.1 Contract type

In lieu of unspent outputs, Tezos uses stateful accounts. When those accounts specify executable code, they are known more generally as contracts. Since an

⁴A zero-knowledge proof would allow anyone to benefit from denouncing malfeasances, but it's not particularly clear this carries much benefit.

account is a type of contract (one with no executable code), we refer to both as “contracts” in full generality.

Each contract has a “manager”, which in the case of an account is simply the owner. If the contract is flagged as spendable, the manager may spend the funds associated with the contract. In addition, each contract may specify the hash of a public key used to sign or mine blocks in the proof-of-stake protocol. The private key may or may not be controlled by the manager.

Formally, a contract is represented as:

```
type contract = {
    counter: int; (* counter to prevent repeat attacks *)
    manager: id; (* hash of the contract's manager public key *)
    balance: Int64.t; (* balance held *)
    signer: id option; (* id of the signer *)
    code: opcode list; (* contract code as a list of opcodes *)
    storage: data list; (* storage of the contract *)
    spendable: bool; (* may the money be spent by the manager? *)
    delegatable: bool; (* may the manager change the signing key? *)
}
```

The handle of a contract is the hash of its initial content. Attempting to create a contract whose hash would collide with an existing contract is an invalid operation and cannot be included in a valid block.

Note that data is represented as the union type.

```
type data =
| STRING of string
| INT of int
```

where INT is a signed 64-bit integer and string is an array of up to 1024 bytes. The storage capacity is limited to 16 384 bytes, counting the integers as eight bytes and the strings as their length.

3.3.2 Origination

The origination operation may be used to create a new contract, it specifies the code of the contract and the initial content of the contract’s storage. If the handle is already the handle of an existing contract, the origination is rejected (there is no reason for this to ever happen, unless by mistake or malice).

A contract needs a minimum balance of 1 to remain active. If the balance falls below this number, the contract is destroyed.

3.3.3 Transactions

A transaction is a message sent from one contract to another contract, this messages is represented as:

```
type transaction = {
    amount: amount; (* amount being sent *)
    parameters: data list; (* parameters passed to the script *)
    (* counter (invoice id) to avoid repeat attacks *)
    counter: int;
    destination: contract hash;
}
```

Such a transaction can be sent from a contract if signed using the manager’s key or can be sent programmatically by code executing in the contract. When the transaction is received, the amount is added to the destination contract’s balance and the destination contract’s code is executed. This code can make use of the parameters passed to it, it can read and write the contract’s storage, change the signature key and post transactions to other contracts.

The role of the counter is to prevent replay attacks. A transaction is only valid if the contract’s counter is equal to the transaction’s counter. Once a transaction is applied, the counter increases by one, preventing the transaction from being reused.

The transaction also includes the block hash of a recent block that the client considers valid. If an attacker ever succeeds in forcing a long reorganization with a fork, he will be unable to include such transactions, making the fork obviously fake. This is a last line of defense, TAPOS is a great system to prevent long reorganizations but not a very good system to prevent short term double spending.

The pair (`account_handle`, `counter`) is roughly the equivalent of an unspent output in Bitcoin.

3.3.4 Storage fees

Since storage imposes a cost on the network, a minimum fee of $\text{\$} 1$ is assessed for each byte increase in the storage. For instance, if after the execution of a transaction, an integer has been added to the storage and ten characters have been appended to an existing string in the storage, then $\text{\$} 18$ will be withdrawn from the contract’s balance and destroyed.

3.3.5 Code

The language is stack based, with high level data types and primitives and strict static type checking. Its design is inspired by Forth, Scheme, ML and Cat. A full specification of the instruction set is available in[5]. This specification gives the complete instruction set, type system and semantics of the language. It is meant as a precise reference manual, not an easy introduction.

3.3.6 Fees

So far, this system is similar to the way Ethereum handles transaction. However, we differ in the way we handle fees. Ethereum allows arbitrarily long programs to execute by requiring a fee that increases linearly with the program’s executing time. Unfortunately, while this does provide an incentive for one miner to verify the transaction, it does not provide such an incentive to other miners, who must also verify this transaction. In practice, most of the interesting programs that can be used for smart contracts are very short. Thus, we simplify the construction by imposing a hard cap on the number of steps we allow the programs to run for.

If the hard cap proves too tight for some programs, they can break the execution in multiple steps and use multiple transactions to execute fully. Since Tezos is amendable, this cap can be changed in the future, or advanced primitives can be introduced as new opcodes.

If the account permits, the signature key may be changed by issuing a signed message requesting the change.

4 Conclusion

We feel we've built an appealing seed protocol. However, Tezos's true potential lies in putting the stakeholders in charge of deciding on a protocol that they feel best serves them.

References

- [1] Vitalik Buterin. Slasher: A punitive proof-of-stake algorithm. <https://blog.ethereum.org/2014/01/15/slasher-a-punitive-proof-of-stake-algorithm/>, 2014.
- [2] Ariel Gabizon Iddo Bentov and Alex Mizrahi. Cryptocurrencies without proof of work. <http://www.cs.technion.ac.il/~iddo/CoA.pdf>, 2014.
- [3] Peter Suber. Nomic: A game of self-amendment. <http://legacy.earlham.edu/~peters/writing/nomic.htm>, 1982.
- [4] Jérôme Vouillon. Lwt: a cooperative thread library. 2008.
- [5] Tezos project. Formal specification of the tezos smart contract language. <https://tezos.com/pages/tech.html>, 2014.

Zerocash: Decentralized Anonymous Payments from Bitcoin (extended version)

Eli Ben-Sasson* Alessandro Chiesa† Christina Garman‡ Matthew Green‡
Ian Miers‡ Eran Tromer§ Madars Virza†

May 18, 2014

Abstract

Bitcoin is the first digital currency to see widespread adoption. Although payments are conducted between pseudonyms, Bitcoin cannot offer strong privacy guarantees: payment transactions are recorded in a public decentralized ledger, from which much information can be deduced. Zerocoin (Miers et al., IEEE S&P 2013) tackles some of these privacy issues by unlinking transactions from the payment’s origin. Yet it still reveals payment destinations and amounts, and is limited in functionality.

In this paper, we construct a full-fledged ledger-based digital currency with strong privacy guarantees. Our results leverage recent advances in *zero-knowledge Succinct Non-interactive ARguments of Knowledge* (zk-SNARKs).

We formulate and construct *decentralized anonymous payment schemes* (DAP schemes). A DAP scheme lets users pay each other directly and privately: the corresponding transaction hides the payment’s origin, destination, and amount. We provide formal definitions and proofs of the construction’s security.

We then build Zerocash, a practical instantiation of our DAP scheme construction. In Zerocash, transactions are less than 1 kB and take under 6 ms to verify — orders of magnitude more efficient than the less-anonymous Zerocoin and competitive with plain Bitcoin.

Keywords: Bitcoin, decentralized electronic cash, zero-knowledge proofs

*Technion, eli@cs.technion.ac.il

†MIT, {alexch, madars}@mit.edu

‡Johns Hopkins University, {cgarman, imiers, mgreen}@cs.jhu.edu

§Tel Aviv University, tromer@cs.tau.ac.il

Contents

1	Introduction	3
1.1	zk-SNARKs	4
1.2	Centralized anonymous payment systems	5
1.3	Decentralized anonymous payment schemes	5
1.4	Zerocash	9
1.5	Paper organization	10
2	Background on zk-SNARKs	10
2.1	Informal definition	10
2.2	Comparison with NIZKs	11
2.3	Known constructions and security	12
2.4	zk-SNARK implementations	12
3	Definition of a decentralized anonymous payment scheme	13
3.1	Data structures	13
3.2	Algorithms	14
3.3	Completeness	16
3.4	Security	16
4	Construction of a decentralized anonymous payment scheme	18
4.1	Cryptographic building blocks	18
4.2	zk-SNARKs for pouring coins	19
4.3	Algorithm constructions	20
4.4	Completeness and security	20
5	Zerocash	20
5.1	Instantiation of building blocks	22
5.2	Arithmetic circuit for pouring coins	23
6	Integration with existing ledger-based currencies	26
6.1	Integration by replacing the base currency	26
6.2	Integration by hybrid currency	26
6.3	Extending the Bitcoin protocol to support the combined semantics	28
6.4	Additional anonymity considerations	28
7	Experiments	28
7.1	Performance of zk-SNARKs for pouring coins	29
7.2	Performance of Zerocash algorithms	29
7.3	Large-scale network simulation	30
8	Optimizations and extensions	33
8.1	Everlasting anonymity	33
8.2	Fast block propagation	34
8.3	Improved storage requirements	34
9	Concurrent work	36
10	Conclusion	36
Acknowledgments		37
A	Overview of Bitcoin and Zerocoin	38
A.1	Bitcoin	38
A.2	Zerocoin	38
B	Completeness of DAP schemes	39
C	Security of DAP schemes	40
C.1	Ledger indistinguishability	41
C.2	Transaction non-malleability	42
C.3	Balance	43
D	Proof of Theorem 4.1	44
D.1	Proof of ledger indistinguishability	44
D.2	Proof of transaction non-malleability	48
D.3	Proof of balance	51
References		54

1 Introduction

Bitcoin is the first digital currency to achieve widespread adoption. The currency owes its rise in part to the fact that, unlike traditional e-cash schemes [Cha82, CHL05, ST99], it requires no trusted parties. Instead of appointing a central bank, Bitcoin uses a distributed ledger known as the *block chain* to store transactions carried out between users. Because the block chain is massively replicated by mutually-distrustful peers, the information it contains is public.

While users may employ many identities (or *pseudonyms*) to enhance their privacy, an increasing body of research shows that anyone can *de-anonymize* Bitcoin by using information in the block chain [RM11, BBSU12, RS12, MPJ⁺13], such as the structure of the transaction graph as well as the value and dates of transactions. As a result, Bitcoin fails to offer even a modicum of the privacy provided by traditional payment systems, let alone the robust privacy of anonymous e-cash schemes.

While Bitcoin is not anonymous itself, those with sufficient motivation can obfuscate their transaction history with the help of *mixes* (also known as *laundries* or *tumblers*). A mix allows users to entrust a set of coins to a pool operated by a central party and then, after some interval, retrieve different coins (with the same total value) from the pool. However, mixes suffer from three limitations: (i) the delay to reclaim coins must be large to allow enough coins to be mixed in; (ii) the mix operator can trace coins; and (iii) the mix operator may steal coins.¹ For users with “something to hide”, these risks may be acceptable. But typical legitimate users (1) wish to keep their spending habits private from their peers, (2) are risk-averse and do not wish to expend continual effort in protecting their privacy, and (3) are often not sufficiently aware that their privacy has been compromised.

To protect their *privacy*, users thus need an instant, risk-free, and, most importantly, automatic guarantee that data revealing their spending habits and account balances is not publicly accessible by their neighbors, co-workers, and the merchants with whom they do business. Anonymous transactions also ensure that the market value of a coin is independent of its history, thus ensuring that legitimate users’ coins remain *fungible*.²

Zerocoins: a decentralized mix. Miers et al. [MGGR13] proposed Zerocoins, which extends Bitcoin to provide strong anonymity guarantees. Like many e-cash protocols (e.g., [CHL05]), Zerocoins employs zero-knowledge proofs to prevent transaction graph analyses. Unlike earlier practical e-cash protocols, however, Zerocoins does not rely on digital signatures to validate coins, nor does it require a central bank to prevent double spending. Instead, Zerocoins authenticates coins by proving, in zero-knowledge, that they belong to a public list of valid coins (which can be maintained on the block chain). Yet rather than a full-fledged anonymous currency, Zerocoins is a *decentralized mix*, where users may periodically “wash” their bitcoins via the Zerocoins protocol. Routine day-to-day transactions must be conducted via Bitcoin, due to reasons that we now review.

The first reason is performance. Redeeming zerocoins requires double-discrete-logarithm proofs of knowledge, which have size that exceeds 45 kB and require 450 ms to verify (at the 128-bit security level).³ These proofs must be broadcast through the network, verified by every node, and permanently stored in the ledger. The entailed costs are higher, by orders of magnitude, than those in Bitcoin and can seriously tax a Bitcoin network operating at normal scale.

¹CoinJoin [Max13], an alternative proposal, replaces the central party of a mix with multi-signature transactions that involve many collaborating Bitcoin users. CoinJoin can thus only mix small volumes of coins amongst users who are currently online, is prone to denial-of-service attacks by third parties, and requires effort to find mixing partners.

²While the methods we detail in this paper accomplish this, the same techniques open the door for privacy-preserving accountability and oversight (see Section 10).

³These published numbers [MGGR13] actually use a mix of parameters at both 128-bit and 80-bit security for different components of the construction. The cost is higher if all parameters are instantiated at 128-bit security.

The second reason is functionality. While Zerocoin constitutes a basic e-cash scheme, it lacks critical features required of full-fledged anonymous payments. First, Zerocoin uses coins of fixed denomination: it does not support payments of exact values, nor does it provide a means to give change following a transaction (i.e., divide coins). Second, Zerocoin has no mechanism for one user to pay another one directly in “zerocoins”. And third, while Zerocoin provides anonymity by unlinking a payment transaction from its origin address, it does not hide the amount or other metadata about transactions occurring on the network.

Our contribution. Addressing this challenge, this work offers two main contributions.

(1) We introduce the notion of a *decentralized anonymous payment scheme*, which formally captures the functionality and security guarantees of a full-fledged decentralized electronic currency with strong anonymity guarantees. We provide a construction of this primitive and prove its security under specific cryptographic assumptions. The construction leverages recent advances in the area of zero-knowledge proofs. Specifically, it uses *zero-knowledge Succinct Non-interactive ARguments of Knowledge* (zk-SNARKs) [Gro10, Lip12, BCI⁺13, GGPR13, PGHR13, BCG⁺13, Lip13, BCTV14].

(2) We implement the above primitive, via a system that we call **Zerocash**. Our system (at 128 bits of security):

- reduces the size of transactions spending a coin to under 1 kB (an improvement of over 97.7%);
- reduces the spend-transaction verification time to under 6 ms (an improvement of over 98.6%);
- allows for anonymous transactions of variable amounts;
- hides transaction amounts and the values of coins held by users; and
- allows for payments to be made directly to a user’s fixed address (without user interaction).

To validate our system, we measured its performance and established feasibility by conducting experiments in a test network of 1000 nodes (approximately $\frac{1}{16}$ of the unique IPs in the Bitcoin network and $\frac{1}{3}$ of the nodes reachable at any given time [DW13]). This inspires confidence that Zerocash can be deployed as a fork of Bitcoin and operate at the same scale. Thus, due to its substantially improved functionality and performance, Zerocash makes it possible to entirely replace traditional Bitcoin payments with anonymous alternatives.

Concurrent work. The idea of using zk-SNARKs in the Bitcoin setting was first presented by one of the authors at Bitcoin 2013 [Ben13]. In concurrent work, Danezis et al. [DFKP13] suggest using zk-SNARKs to reduce proof size and verification time in Zerocoin; see Section 9 for a comparison.

1.1 zk-SNARKs

A zk-SNARK is an efficient variant of a *zero-knowledge proof of knowledge* [GMR89], which we first informally describe via an example. Suppose Alice wishes to prove to Bob the statement “*I (Alice) own 30 bitcoins*”. A simple method for Alice to do so is to point to 30 coins on the block chain and, for each of them, sign a message (“hello, world”) using the secret key that controls that coin. Alas, this method *leaks knowledge* to Bob, by identifying which coins are Alice’s. A zero-knowledge proof of knowledge allows Alice to achieve the same goal, while revealing *no information* to Bob (beyond the fact that she *knows* some secret keys that control 30 coins). Crucially, such proofs can be obtained for any statement that can be verified to be true by use of an efficient computation involving auxiliary inputs such as trapdoors and passwords (such statements are called “NP statements”).

We now sketch in more technical terms the definition of a zk-SNARK; see Section 2 for more details. A zk-SNARK is a non-interactive zero-knowledge proof of knowledge that is *succinct*, i.e., for which proofs are very short and easy to verify. More precisely, let \mathcal{L} be an NP language, and let C be a nondeterministic decision circuit for \mathcal{L} on a given instance size n . A zk-SNARK can be used

to prove and verify membership in \mathcal{L} , for instances of size n , as follows. After taking C as input, a trusted party conducts a one-time setup phase that results in two public keys: a proving key pk and a verification key vk . The proving key pk enables any (untrusted) prover to produce a proof π attesting to the fact that $x \in \mathcal{L}$, for an instance x (of size n) of his choice. The non-interactive proof π is *zero knowledge* and a *proof of knowledge*. Anyone can use the verification key vk to verify the proof π ; in particular zk-SNARK proofs are publicly verifiable: anyone can verify π , without ever having to interact with the prover who generated π . Succinctness requires that (for a given security level) π has *constant size* and can be verified in time that is linear in $|x|$ (rather than linear in $|C|$).

1.2 Centralized anonymous payment systems

Before describing our new decentralized payment system, we put it in context by recalling two pre-Bitcoin payment schemes, both of which relied on a *bank*, acting as a central trusted party.

Anonymous e-cash. Chaum [Cha82] first obtained anonymous e-cash. In Chaum’s scheme, the minting of a coin involves both a user, Alice, and the bank: to mint a coin of a given value v , Alice first selects a random secret serial number sn (unknown to the bank); then, the bank, after deducting v from Alice’s balance, signs sn via a *blind signature*. Afterwards, if Alice wants to transfer her coin to Bob, she reveals sn to him and proves that sn was signed by the bank; during this transfer, Bob (or the bank) cannot deduce Alice’s identity from the revealed information. Double-spending is prevented because the bank will not honor a coin with a previously-seen serial number.

Unforgeable e-cash. One problem with Chaum’s scheme is that coins can be forged if the bank’s secret key is compromised. Sander and Ta-Shma [ST99] addressed this, as follows. The bank maintains a public Merkle tree of “coin commitments”, and users periodically retrieve its root rt ; in particular, the bank maintains no secrets. When Alice requests a coin (of unit value), she picks a random serial number sn and auxiliary string r , and then sends $\text{cm} := \text{CRH}(\text{sn}\|r)$ to the bank, where CRH is a collision-resistant hash; the bank deducts the appropriate amount from Alice’s balance and then records cm as a leaf in the Merkle tree. Afterwards, to pay Bob, Alice sends him sn along with a zero-knowledge proof of knowledge π of the following NP statement: “*there exists r such that $\text{CRH}(\text{sn}\|r)$ is a leaf in a Merkle tree with root rt* ”. In other words, Alice can convince Bob that sn is the serial number contained in *some* coin commitment in the Merkle tree; but the zero-knowledge property prevents Bob from learning information about which coin commitment is Alice’s, thereby protecting Alice’s identity. Later, Bob can “cash out” Alice’s coin by showing sn and π to the bank.⁴

Moving to a fungible anonymous decentralized system. In this paper, like [ST99], we hash a coin’s serial number and use Merkle trees to compactly represent the set of minted coins. Unlike [ST99], we also ensure the privacy of a coin’s value and support transactions that split and merge coins, thus achieving (and implementing) a new kind of fully-fungible and divisible payment scheme. As in Bitcoin (and in stark contrast to previous e-cash schemes), we do not rely on a trusted bank. Therefore, we require a new set of definitions and protocols, designed to protect Alice’s anonymity while preventing her from falsely increasing her balance under the veil of her boosted privacy. An informal description of our payment scheme follows.

1.3 Decentralized anonymous payment schemes

We construct a *decentralized anonymous payment (DAP) scheme*, which is a decentralized e-cash scheme that allows direct anonymous payments of any amount. See Section 3 for a formal definition.

⁴We omit details about how the bank can identify Alice in the event that she double spends her coin.

Here, we outline our construction in six incremental steps; the construction details are in Section 4.

Our construction functions on top of any ledger-based base currency, such as Bitcoin. At any given time, a unique valid snapshot of the currency’s *ledger* is available to all users. The ledger is a sequence of *transactions* and is append-only. Transactions include both the underlying currency’s transactions, as well as new transactions introduced by our construction. For concreteness, we focus the discussion below on Bitcoin (though later definitions and constructions are stated abstractly). We assume familiarity with Bitcoin [Nak09] and Zerocoin [MGGR13]; both are reviewed in Appendix A.

Step 1: user anonymity with fixed-value coins. We first describe a simplified construction, in which all coins have the same value of, e.g., 1 BTC. This construction, similar to the Zerocoin protocol, shows how to hide a payment’s origin. In terms of tools, we make use of zk-SNARKs (recalled above) and a commitment scheme. Let COMM denote a statistically-hiding non-interactive commitment scheme (i.e., given randomness r and message m , the commitment is $c := \text{COMM}_r(m)$; subsequently, c is opened by revealing r and m , and one can verify that $\text{COMM}_r(m)$ equals c).

In the simplified construction, a new coin \mathbf{c} is minted as follows: a user u samples a random *serial number* \mathbf{sn} and a *trapdoor* r , computes a *coin commitment* $\mathbf{cm} := \text{COMM}_r(\mathbf{sn})$, and sets $\mathbf{c} := (r, \mathbf{sn}, \mathbf{cm})$. A corresponding mint transaction $\mathbf{tx}_{\text{Mint}}$, containing \mathbf{cm} (but not \mathbf{sn} or r), is sent to the ledger; $\mathbf{tx}_{\text{Mint}}$ is appended to the ledger only if u has paid 1 BTC to a backing escrow pool (e.g., the 1 BTC may be paid via plaintext information encoded in $\mathbf{tx}_{\text{Mint}}$). Mint transactions are thus certificates of deposit, deriving their value from the backing pool.

Subsequently, letting CMList denote the list of all coin commitments on the ledger, u may spend \mathbf{c} by posting a spend transaction $\mathbf{tx}_{\text{Spend}}$ that contains (i) the coin’s serial number \mathbf{sn} ; and (ii) a zk-SNARK proof π of the NP statement “*I know r such that $\text{COMM}_r(\mathbf{sn})$ appears in the list CMList of coin commitments*”. Assuming that \mathbf{sn} does not already appear on the ledger (as part of a past spend transaction), u can redeem the deposited amount of 1 BTC, which u can either keep, transfer to someone else, or mint a new coin. (If \mathbf{sn} does already appear on the ledger, this is considered double spending, and the transaction is discarded.)

User anonymity is achieved because the proof π is zero-knowledge: while \mathbf{sn} is revealed, no information about r is, and finding which of the numerous commitments in CMList corresponds to a particular spend transaction $\mathbf{tx}_{\text{Spend}}$ is equivalent to inverting $f(x) := \text{COMM}_x(\mathbf{sn})$, which is assumed to be infeasible. Thus, the origin of the payment is anonymous.

Step 2: compressing the list of coin commitments. In the above NP statement, CMList is specified explicitly as a list of coin commitments. This naive representation severely limits scalability because the time and space complexity of most protocol algorithms (e.g., the proof verification algorithm) grow linearly with CMList . Moreover, coin commitments corresponding to already-spent coins cannot be dropped from CMList to reduce costs, since they cannot be identified (due to the same zero-knowledge property that provides anonymity).

As in [ST99], we rely on a collision-resistant function CRH to avoid an explicit representation of CMList . We maintain an efficiently-updatable append-only CRH -based Merkle tree $\text{Tree}(\text{CMList})$ over the (growing) list CMList and let \mathbf{rt} denote the root of $\text{Tree}(\text{CMList})$. It is well-known that \mathbf{rt} can be updated to account for the insertion of new leaves with time and space proportional to just the tree depth. Hence, the time and space complexity is reduced from linear in the size of CMList to logarithmic. With this in mind, we modify the NP statement to the following one: “*I know r such that $\text{COMM}_r(\mathbf{sn})$ appears as a leaf in a CRH-based Merkle tree whose root is \mathbf{rt}* ”. Compared with the naive data structure for CMList , this modification increases exponentially the size of CMList that a given zk-SNARK implementation can support. (Concretely: using Merkle trees of depth 64, Zerocash supports 2^{64} coins.)

Step 3: extending coins for direct anonymous payments. So far, the coin commitment

cm of a coin \mathbf{c} is a commitment to the coin's serial number sn . However, this creates a problem when transferring \mathbf{c} to another user. Indeed, suppose that a user u_A created \mathbf{c} , and u_A sends \mathbf{c} to another user u_B . First, since u_A knows sn , the spending of \mathbf{c} by u_B is both non-anonymous (since u_A sees when \mathbf{c} is spent, by recognizing sn) and risky (since u_A could still spend \mathbf{c} first). Thus, u_B must immediately spend \mathbf{c} and mint a new coin \mathbf{c}' to protect himself. Second, if u_A in fact wants to transfer to u_B , e.g., 100 BTC, then doing so is both unwieldy (since it requires 100 transfers) and non-anonymous (since the amount of the transfer is leaked). And third, transfers in amounts that are not multiples of 1 BTC (the fixed value of a coin) are not supported. Thus, the simplified construction described is inadequate as a payment scheme.

We address this by modifying the derivation of a coin commitment, and using pseudorandom functions to target payments and to derive serial numbers, as follows. We use three pseudorandom functions (derived from a single one). For a seed x , these are denoted $\text{PRF}_x^{\text{addr}}(\cdot)$, $\text{PRF}_x^{\text{sn}}(\cdot)$, and $\text{PRF}_x^{\text{pk}}(\cdot)$. We assume that PRF^{sn} is moreover collision-resistant.

To provide targets for payments, we use *addresses*: each user u generates an address key pair $(a_{\text{pk}}, a_{\text{sk}})$, the *address public key* and *address private key* respectively. The coins of u contain the value a_{pk} and can be spent only with knowledge of a_{sk} . A key pair $(a_{\text{pk}}, a_{\text{sk}})$ is sampled by selecting a random seed a_{sk} and setting $a_{\text{pk}} := \text{PRF}_{a_{\text{sk}}}^{\text{addr}}(0)$. A user can generate and use any number of address key pairs.

Next, we redesign minting to allow for greater functionality. To mint a coin \mathbf{c} of a desired value v , the user u first samples ρ , which is a secret value that determines the coin's serial number as $\text{sn} := \text{PRF}_{a_{\text{sk}}}^{\text{sn}}(\rho)$. Then, u commits to the tuple (a_{pk}, v, ρ) in two phases: (a) u computes $k := \text{COMM}_r(a_{\text{pk}} \| \rho)$ for a random r ; and then (b) u computes $\text{cm} := \text{COMM}_s(v \| k)$ for a random s . The minting results in a coin $\mathbf{c} := (a_{\text{pk}}, v, \rho, r, s, \text{cm})$ and a mint transaction $\text{tx}_{\text{Mint}} := (v, k, s, \text{cm})$. Crucially, due to the nested commitment, anyone can verify that cm in tx_{Mint} is a coin commitment of a coin of value v (by checking that $\text{COMM}_s(v \| k)$ equals cm) but cannot discern the owner (by learning the address key a_{pk}) or serial number (derived from ρ) because these are hidden in k . As before, tx_{Mint} is accepted by the ledger only if u deposits the correct amount, in this case v BTC.

Coin are spent using the *pour* operation, which takes a set of input coins, to be consumed, and “pours” their value into a set of fresh output coins — such that the total value of output coins equals the total value of the input coins. Suppose that u , with address key pair $(a_{\text{pk}}^{\text{old}}, a_{\text{sk}}^{\text{old}})$, wishes to consume his coin $\mathbf{c}^{\text{old}} = (a_{\text{pk}}^{\text{old}}, v^{\text{old}}, \rho^{\text{old}}, r^{\text{old}}, s^{\text{old}}, \text{cm}^{\text{old}})$ and produce two new coins $\mathbf{c}_1^{\text{new}}$ and $\mathbf{c}_2^{\text{new}}$, with total value $v_1^{\text{new}} + v_2^{\text{new}} = v^{\text{old}}$, respectively targeted at address public keys $a_{\text{pk},1}^{\text{new}}$ and $a_{\text{pk},2}^{\text{new}}$. (The addresses $a_{\text{pk},1}^{\text{new}}$ and $a_{\text{pk},2}^{\text{new}}$ may belong to u or to some other user.) The user u , for each $i \in \{1, 2\}$, proceeds as follows: (i) u samples serial number randomness ρ_i^{new} ; (ii) u computes $k_i^{\text{new}} := \text{COMM}_{r_i^{\text{new}}}(a_{\text{pk},i}^{\text{new}} \| \rho_i^{\text{new}})$ for a random r_i^{new} ; and (iii) u computes $\text{cm}_i^{\text{new}} := \text{COMM}_{s_i^{\text{new}}}(v_i^{\text{new}} \| k_i^{\text{new}})$ for a random s_i^{new} .

This yields the coins $\mathbf{c}_1^{\text{new}} := (a_{\text{pk},1}^{\text{new}}, v_1^{\text{new}}, \rho_1^{\text{new}}, r_1^{\text{new}}, s_1^{\text{new}}, \text{cm}_1^{\text{new}})$ and $\mathbf{c}_2^{\text{new}} := (a_{\text{pk},2}^{\text{new}}, v_2^{\text{new}}, \rho_2^{\text{new}}, r_2^{\text{new}}, s_2^{\text{new}}, \text{cm}_2^{\text{new}})$. Next, u produces a zk-SNARK proof π_{POUR} for the following NP statement, which we call **POUR**:

“Given the Merkle-tree root rt , serial number sn^{old} , and coin commitments $\text{cm}_1^{\text{new}}, \text{cm}_2^{\text{new}}$, I know coins $\mathbf{c}^{\text{old}}, \mathbf{c}_1^{\text{new}}, \mathbf{c}_2^{\text{new}}$, and address secret key $a_{\text{sk}}^{\text{old}}$ such that:

- The coins are well-formed: for \mathbf{c}^{old} it holds that $k^{\text{old}} = \text{COMM}_{r^{\text{old}}}(a_{\text{pk}}^{\text{old}} \| \rho^{\text{old}})$ and $\text{cm}^{\text{old}} = \text{COMM}_{s^{\text{old}}}(v^{\text{old}} \| k^{\text{old}})$; and similarly for $\mathbf{c}_1^{\text{new}}$ and $\mathbf{c}_2^{\text{new}}$.
- The address secret key matches the public key: $a_{\text{pk}}^{\text{old}} = \text{PRF}_{a_{\text{sk}}^{\text{old}}}^{\text{addr}}(0)$.
- The serial number is computed correctly: $\text{sn}^{\text{old}} := \text{PRF}_{a_{\text{sk}}^{\text{old}}}^{\text{sn}}(\rho^{\text{old}})$.
- The coin commitment cm^{old} appears as a leaf of a Merkle-tree with root rt .
- The values add up: $v_1^{\text{new}} + v_2^{\text{new}} = v^{\text{old}}$.”

A resulting pour transaction $\text{tx}_{\text{Pour}} := (\text{rt}, \text{sn}^{\text{old}}, \text{cm}_1^{\text{new}}, \text{cm}_2^{\text{new}}, \pi_{\text{POUR}})$ is appended to the ledger. (As before, the transaction is rejected if the serial number sn appears in a previous transaction.)

Now suppose that u does not know, say, the address secret key $a_{\text{sk},1}^{\text{new}}$ that is associated with the public key $a_{\text{pk},1}^{\text{new}}$. Then, u cannot spend $\mathbf{c}_1^{\text{new}}$ because he cannot provide $a_{\text{sk},1}^{\text{new}}$ as part of the witness of a subsequent pour operation. Furthermore, when a user who knows $a_{\text{sk},1}^{\text{new}}$ does spend $\mathbf{c}_1^{\text{new}}$, the user u cannot track it, because he knows no information about its revealed serial number, which is $\text{sn}_1^{\text{new}} := \text{PRF}_{a_{\text{sk},1}^{\text{new}}}^{\text{sn}}(\rho_1^{\text{new}})$.

Also observe that tx_{Pour} reveals no information about how the value of the consumed coin was divided among the two new fresh coins, nor which coin commitment corresponds to the consumed coin, nor the address public keys to which the two new fresh coins are targeted. The payment was conducted in full anonymity.

More generally, a user may pour $N^{\text{old}} \geq 0$ coins into $N^{\text{new}} \geq 0$ coins. For simplicity we consider the case $N^{\text{old}} = N^{\text{new}} = 2$, without loss of generality. Indeed, for $N^{\text{old}} < 2$, the user can mint a coin with value 0 and then provide it as a “null” input, and for $N^{\text{new}} < 2$, the user can create (and discard) a new coin with value 0. For $N^{\text{old}} > 2$ or $N^{\text{new}} > 2$, the user can compose $\log N^{\text{old}} + \log N^{\text{new}}$ of the 2-input/2-output pours.

Step 4: sending coins. Suppose that $a_{\text{pk},1}^{\text{new}}$ is the address public key of u_1 . In order to allow u_1 to actually spend the new coin $\mathbf{c}_1^{\text{new}}$ produced above, u must somehow send the secret values in $\mathbf{c}_1^{\text{new}}$ to u_1 . One way is for u to send u_1 a private message, but the requisite private communication channel necessitates additional infrastructure or assumptions. We avoid this “out-of-band” channel and instead build this capability directly into our construction by leveraging the ledger as follows.

We modify the structure of an address key pair. Each user now has a key pair $(\text{addr}_{\text{pk}}, \text{addr}_{\text{sk}})$, where $\text{addr}_{\text{pk}} = (a_{\text{pk}}, \text{pk}_{\text{enc}})$ and $\text{addr}_{\text{sk}} = (a_{\text{sk}}, \text{sk}_{\text{enc}})$. The values $(a_{\text{pk}}, a_{\text{sk}})$ are generated as before. In addition, $(\text{pk}_{\text{enc}}, \text{sk}_{\text{enc}})$ is a key pair for a *key-private encryption scheme* [BBDP01].

Then, u computes the ciphertext \mathbf{C}_1 that is the encryption of the plaintext $(v_1^{\text{new}}, \rho_1^{\text{new}}, r_1^{\text{new}}, s_1^{\text{new}})$, under $\text{pk}_{\text{enc},1}^{\text{new}}$ (which is part of u_1 ’s address public key $\text{addr}_{\text{sk},1}^{\text{new}}$), and includes \mathbf{C}_1 in the pour transaction tx_{Pour} . The user u_1 can then find and decrypt this message (using his $\text{sk}_{\text{enc},1}^{\text{new}}$) by scanning the pour transactions on the public ledger. Again, note that adding \mathbf{C}_1 to tx_{Pour} leaks neither paid amounts, nor target addresses due to the key-private property of the encryption scheme. (The user u does the same with $\mathbf{c}_2^{\text{new}}$ and includes a corresponding ciphertext \mathbf{C}_2 in tx_{Pour} .)

Step 5: public outputs. The construction so far allows users to mint, merge, and split coins. But how can a user redeem one of his coins, i.e., convert it back to the base currency (Bitcoin)? For this, we modify the pour operation to include a *public output*. When spending a coin, the user u also specifies a nonnegative v_{pub} and a *transaction string* $\text{info} \in \{0,1\}^*$. The balance equation in the NP statement POUR is changed accordingly: “ $v_1^{\text{new}} + v_2^{\text{new}} + v_{\text{pub}} = v^{\text{old}}$ ”. Thus, of the input value v^{old} , a part v_{pub} is publicly declared, and its target is specified, somehow, by the string info . The string info can be used to specify the destination of these redeemed funds (e.g., a Bitcoin wallet public key).⁵ Both v_{pub} and info are now included in the resulting pour transaction tx_{Pour} . (The public output is optional, as the user u can set $v_{\text{pub}} = 0$.)

Step 6: non-malleability. To prevent malleability attacks on a pour transaction tx_{Pour} (e.g., embezzlement by re-targeting the public output of the pour by modifying info), we further modify the NP statement POUR and use digital signatures. Specifically, during the pour operation, the user u (i) samples a key pair $(\text{pk}_{\text{sig}}, \text{sk}_{\text{sig}})$ for a one-time signature scheme; (ii) computes $h_{\text{Sig}} := \text{CRH}(\text{pk}_{\text{sig}})$; (iii) computes the two values $h_1 := \text{PRF}_{a_{\text{sk},1}^{\text{old}}}^{\text{pk}}(h_{\text{Sig}})$ and $h_2 := \text{PRF}_{a_{\text{sk},2}^{\text{old}}}^{\text{pk}}(h_{\text{Sig}})$, which act as MACs to

⁵These public outputs can be considered as an “input” to a Bitcoin-style transaction, where the string info contains the Bitcoin output scripts. This mechanism also allows us to support Bitcoin’s public transaction fees.

“tie” h_{Sig} to both address secret keys; (iv) modifies POUR to include the three values h_{Sig}, h_1, h_2 and prove that the latter two are computed correctly; and (v) uses sk_{sig} to sign every value associated with the pour operation, thus obtaining a signature σ , which is included, along with pk_{sig} , in tx_{Pour} . Since the $a_{\text{sk},i}^{\text{old}}$ are secret, and with high probability h_{Sig} changes for each pour transaction, the values h_1, h_2 are unpredictable. Moreover, the signature on the NP statement (and other values) binds all of these together, as argued in more detail in Appendix C and Appendix D.

This ends the outline of the construction, which is summarized in part in Figure 1. We conclude by noting that, due to the zk-SNARK, our construction requires a one-time trusted setup of public parameters. The soundness of the proofs depends on this trust, though anonymity continues to hold even if the setup is corrupted by a malicious party.

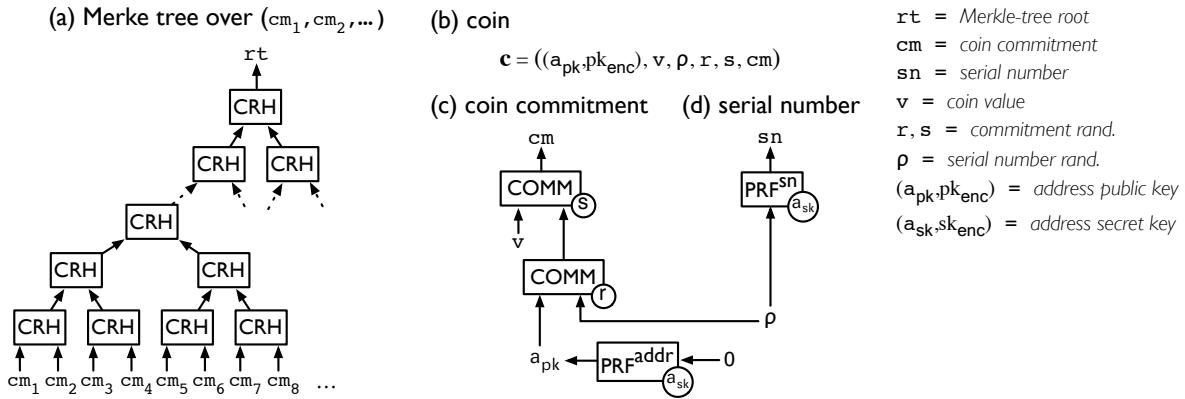


Figure 1: (a) Illustration of the CRH-based Merkle tree over the list CMList of coin commitments. (b) A coin c . (c) Illustration of the structure of a coin commitment cm . (d) Illustration of the structure of a coin serial number sn .

1.4 Zerocash

We outline Zerocash, a concrete implementation, at 128 bits of security, of our DAP scheme construction; see Section 5 for details. Zerocash entails carefully instantiating the cryptographic ingredients of the construction to ensure that the zk-SNARK, the “heaviest” component, is efficient enough in practice. In the construction, the zk-SNARK is used to prove/verify a specific NP statement: POUR . While zk-SNARKs are asymptotically efficient, their concrete efficiency depends on the arithmetic circuit C that is used to decide the NP statement. Thus, we seek instantiations for which we can design a relatively small arithmetic circuit C_{POUR} for verifying the NP statement POUR .

Our approach is to instantiate all of the necessary cryptographic ingredients (commitment schemes, pseudorandom functions, and collision-resistant hashing) based on SHA256. We first design a hand-optimized circuit for verifying SHA256 computations (or, more precisely, its compression function, which suffices for our purposes).⁶ Then, we use this circuit to construct C_{POUR} , which verifies all the necessary checks for satisfying the NP statement C_{POUR} .

This, along with judicious parameter choices, and a state-of-the-art implementation of a zk-SNARK for arithmetic circuits [BCTV14] (see Section 2.4), results in a zk-SNARK prover

⁶ Alternatively, we could have opted to rely on the circuit generators [PGHR13, BCG⁺13, BCTV14], which support various classes of C programs, by writing C code expressing the POUR checks. However, as discussed later, these generic approaches are more expensive than our hand-optimized construction.

running time of a few minutes and zk-SNARK verifier running time of a few milliseconds. This allows the DAP scheme implementation to be practical for deployment, as our experiments show.

Zerocash can be integrated into Bitcoin or forks of it (commonly referred to as “altcoins”); we later describe how this is done.

1.5 Paper organization

The remainder of this paper is organized as follows. Section 2 provides background on zk-SNARKs. We define DAP schemes in Section 3, and our construction thereof in Section 4. Section 5 discusses the concrete instantiation in Zerocash. Section 6 describes the integration of Zerocash into existing ledger-based currencies. Section 7 provides microbenchmarks for our prototype implementation, as well as results based on full-network simulations. Section 8 describes optimizations. We discuss concurrent work in Section 9 and summarize our contributions and future directions in Section 10.

2 Background on zk-SNARKs

The main cryptographic primitive used in this paper is a special kind of *Succinct Non-interactive Argument of Knowledge* (SNARK). Concretely, we use a *publicly-verifiable preprocessing zero-knowledge* SNARK, or zk-SNARK for short. In this section we provide basic background on zk-SNARKs, provide an informal definition, compare zk-SNARKs with the more familiar notion of NIZKs, and recall known constructions and implementations.

2.1 Informal definition

We informally define zk-SNARKs for arithmetic circuit satisfiability. We refer the reader to, e.g., [BCI⁺13] for a formal definition.

For a field \mathbb{F} , an \mathbb{F} -arithmetic circuit takes inputs that are elements in \mathbb{F} , and its gates output elements in \mathbb{F} . We naturally associate a circuit with the function it computes. To model non-determinism we consider circuits that have an *input* $x \in \mathbb{F}^n$ and an auxiliary input $a \in \mathbb{F}^h$, called a *witness*. The circuits we consider only have *bilinear gates*.⁷ Arithmetic circuit satisfiability is defined analogously to the boolean case, as follows.

Definition 2.1. The *arithmetic circuit satisfiability problem* of an \mathbb{F} -arithmetic circuit $C: \mathbb{F}^n \times \mathbb{F}^h \rightarrow \mathbb{F}^l$ is captured by the relation $\mathcal{R}_C = \{(x, a) \in \mathbb{F}^n \times \mathbb{F}^h : C(x, a) = 0^l\}$; its language is $\mathcal{L}_C = \{x \in \mathbb{F}^n : \exists a \in \mathbb{F}^h \text{ s.t. } C(x, a) = 0^l\}$.

Given a field \mathbb{F} , a (publicly-verifiable preprocessing) **zk-SNARK** for \mathbb{F} -arithmetic circuit satisfiability is a triple of polynomial-time algorithms (**KeyGen**, **Prove**, **Verify**):

- **KeyGen**($1^\lambda, C$) $\rightarrow (\mathbf{pk}, \mathbf{vk})$. On input a security parameter λ (presented in unary) and an \mathbb{F} -arithmetic circuit C , the *key generator* **KeyGen** probabilistically samples a *proving key* \mathbf{pk} and a *verification key* \mathbf{vk} . Both keys are published as public parameters and can be used, any number of times, to prove/verify membership in \mathcal{L}_C .
- **Prove**(\mathbf{pk}, x, a) $\rightarrow \pi$. On input a proving key \mathbf{pk} and any $(x, a) \in \mathcal{R}_C$, the *prover* **Prove** outputs a non-interactive proof π for the statement $x \in \mathcal{L}_C$.

⁷A gate with inputs $y_1, \dots, y_m \in \mathbb{F}$ is *bilinear* if the output is $\langle \vec{a}, (1, y_1, \dots, y_m) \rangle \cdot \langle \vec{b}, (1, y_1, \dots, y_m) \rangle$ for some $\vec{a}, \vec{b} \in \mathbb{F}^{m+1}$. These include addition, multiplication, negation, and constant gates.

- $\text{Verify}(\text{vk}, x, \pi) \rightarrow b$. On input a verification key vk , an input x , and a proof π , the *verifier* Verify outputs $b = 1$ if he is convinced that $x \in \mathcal{L}_C$.

A zk-SNARK satisfies the following properties.

Completeness. For every security parameter λ , any \mathbb{F} -arithmetic circuit C , and any $(x, a) \in \mathcal{R}_C$, the honest prover can convince the verifier. Namely, $b = 1$ with probability $1 - \text{negl}(\lambda)$ in the following experiment: $(\text{pk}, \text{vk}) \leftarrow \text{KeyGen}(1^\lambda, C)$; $\pi \leftarrow \text{Prove}(\text{pk}, x, a)$; $b \leftarrow \text{Verify}(\text{vk}, x, \pi)$.

Succinctness. An honestly-generated proof π has $O_\lambda(1)$ bits and $\text{Verify}(\text{vk}, x, \pi)$ runs in time $O_\lambda(|x|)$. (Here, O_λ hides a fixed polynomial factor in λ .)

Proof of knowledge (and soundness). If the verifier accepts a proof output by a bounded prover, then the prover “knows” a witness for the given instance. (In particular, soundness holds against bounded provers.) Namely, for every $\text{poly}(\lambda)$ -size adversary \mathcal{A} , there is a $\text{poly}(\lambda)$ -size extractor \mathcal{E} such that $\text{Verify}(\text{vk}, x, \pi) = 1$ and $(x, a) \notin \mathcal{R}_C$ with probability $\text{negl}(\lambda)$ in the following experiment: $(\text{pk}, \text{vk}) \leftarrow \text{KeyGen}(1^\lambda, C)$; $(x, \pi) \leftarrow \mathcal{A}(\text{pk}, \text{vk})$; $a \leftarrow \mathcal{E}(\text{pk}, \text{vk})$.

Perfect zero knowledge. An honestly-generated proof is perfect zero knowledge.⁸ Namely, there is a polynomial-time simulator Sim such that for all stateful distinguishers \mathcal{D} the following two probabilities are equal:

$$\Pr \left[\begin{array}{c} (x, a) \in \mathcal{R}_C \\ \mathcal{D}(\pi) = 1 \end{array} \middle| \begin{array}{c} (\text{pk}, \text{vk}) \leftarrow \text{KeyGen}(1^\lambda, C) \\ (x, a) \leftarrow \mathcal{D}(\text{pk}, \text{vk}) \\ \pi \leftarrow \text{Prove}(\text{pk}, x, a) \end{array} \right] \text{ and } \Pr \left[\begin{array}{c} (x, a) \in \mathcal{R}_C \\ \mathcal{D}(\pi) = 1 \end{array} \middle| \begin{array}{c} (\text{pk}, \text{vk}, \text{trap}) \leftarrow \text{Sim}(1^\lambda, C) \\ (x, a) \leftarrow \mathcal{D}(\text{pk}, \text{vk}) \\ \pi \leftarrow \text{Sim}(\text{trap}, x) \end{array} \right] .$$

(the probability that $\mathcal{D}(\pi) = 1$ on an honest proof) (the probability that $\mathcal{D}(\pi) = 1$ on a simulated proof)

Remark. Both proof of knowledge and zero knowledge are essential to the use of zk-SNARKs in this paper. Indeed, we consider circuits C that verify assertions about cryptographic primitives (such as using a knowledge of SHA256 pre-image as a binding commitment). Thus it *does not suffice* to merely know that, for a given input x , a witness for $x \in \mathcal{L}_C$ exists. Instead, proof of knowledge ensures that a witness can be efficiently found (by extracting it from the prover) whenever the verifier accepts a proof. As for zero knowledge, it ensures that a proof leaks no information about the witness, beyond the fact that $x \in \mathcal{L}_C$.

Remark. In the security proofs (see Appendix D), we deal with provers producing a vector of inputs \vec{x} together with a vector of corresponding proofs $\vec{\pi}$. In such cases, it is convenient to use an extractor that can extract a vector of witnesses \vec{a} containing a valid witness for each valid proof. This “multi-instance” extraction follows from the “single-instance” one described above [BCCT12, BCCT13]. Namely, if $(\text{KeyGen}, \text{Prove}, \text{Verify})$ is a zk-SNARK, then for any $\text{poly}(\lambda)$ -size prover adversary \mathcal{A} there exists a $\text{poly}(\lambda)$ -size extractor \mathcal{E} such that

$$\Pr \left[\exists i \text{ s.t. } \begin{array}{c} \text{Verify}(\text{vk}, x_i, \pi_i) = 1 \\ (x_i, a_i) \notin \mathcal{R}_C \end{array} \middle| \begin{array}{c} (\text{pk}, \text{vk}) \leftarrow \text{KeyGen}(1^\lambda, C) \\ (\vec{x}, \vec{\pi}) \leftarrow \mathcal{A}(\text{pk}, \text{vk}) \\ \vec{a} \leftarrow \mathcal{E}(\text{pk}, \text{vk}) \end{array} \right] \leq \text{negl}(\lambda) .$$

2.2 Comparison with NIZKs

zk-SNARKs are related to a familiar cryptographic primitive: *non-interactive zero-knowledge proofs of knowledge* (NIZKs). Both zk-SNARKs and NIZKs require a one-time trusted setup of public

⁸While most zk-SNARK descriptions in the literature only mention statistical zero knowledge, all zk-SNARK constructions can be made perfect zero knowledge by allowing for a negligible error probability in completeness.

parameters (proving and verification keys for zk-SNARKs, and a common reference string for NIZKs). Both provide the same guarantees of completeness, proof of knowledge, and zero knowledge. The difference lies in efficiency guarantees. In a NIZK, the proof length and verification time depend on the NP language being proved. For instance, for the language of circuit satisfiability, the proof length and verification time in [GOS06b, GOS06a] are linear in the circuit size. Conversely, in a zk-SNARK, proof length depends only on the security parameter, and verification time depends only on the instance size (and security parameter) but not on the circuit or witness size.

Thus, zk-SNARKs can be thought of as “succinct NIZKs”, having short proofs and fast verification times. Succinctness comes with a caveat: known zk-SNARK constructions rely on stronger assumptions than NIZKs do (see below).

2.3 Known constructions and security

There are many zk-SNARK constructions in the literature [Gro10, Lip12, BCI⁺13, GGPR13, PGHR13, BCG⁺13, Lip13, BCTV14]. We are interested in zk-SNARKs for arithmetic circuit satisfiability, and the most efficient ones for this language are based on *quadratic arithmetic programs* [GGPR13, BCI⁺13, PGHR13, BCG⁺13, BCTV14]; such constructions provide a linear-time KeyGen, quasilinear-time Prove, and linear-time Verify.

Security of zk-SNARKs is based on knowledge-of-exponent assumptions and variants of Diffie–Hellman assumptions in bilinear groups [Gro10, BB04, Gen04]. While knowledge-of-exponent assumptions are fairly strong, there is evidence that such assumptions may be inherent for constructing zk-SNARKs [GW11, BCCT12].

Remark (fully-succinct zk-SNARKs). The key generator KeyGen takes a circuit C as input. Thus, KeyGen’s running time is at least linear in the size of the circuit C . One could require KeyGen to *not* have to take C as input, and have its output keys work for *all* (polynomial-size) circuits C . In such a case, KeyGen’s running time would be independent of C . A zk-SNARK satisfying this stronger property is *fully succinct*. Theoretical constructions of fully-succinct zk-SNARKs are known, based on various cryptographic assumptions [Mic00, Val08, BCCT13]. Despite achieving essentially-optimal asymptotics [BFLS91, BGH⁺05, BCGT13b, BCGT13a, BCCT13] no implementations of them have been reported in the literature to date.

2.4 zk-SNARK implementations

There are three published implementations of zk-SNARKs: (i) Parno et al. [PGHR13] present an implementation of zk-SNARKs for programs having no data dependencies;⁹ (ii) Ben-Sasson et al. [BCG⁺13] present an implementation of zk-SNARKs for arbitrary programs (with data dependencies); and (iii) Ben-Sasson et al. [BCTV14] present an implementation of zk-SNARKs that supports programs that modify their own code (e.g., for runtime code generation); their implementation also reduces costs for programs of larger size and allows for universal key pairs.

Each of the works above also achieves zk-SNARKs for arithmetic circuit satisfiability as a stepping stone towards their respective higher-level efforts. In this paper we are only interested in a zk-SNARK for arithmetic circuit satisfiability, and we rely on the implementation of [BCTV14] for such a zk-SNARK.¹⁰ The implementation in [BCTV14] is itself based on the protocol of Parno et al. [PGHR13]. We thus refer the interested reader to [PGHR13] for details of the protocol, its

⁹They only support programs where array indices are restricted to be known compile-time constants; similarly, loop iteration counts (or at least upper bounds to these) must be known at compile time.

¹⁰In [BCTV14], one optimization to the verifier’s runtime requires preprocessing the verification key vk ; for simplicity, we do not use this optimization.

intuition, and its proof of security; and to [BCTV14] for the implementation and its performance. In terms of concrete parameters, the implementation of [BCTV14] provides 128 bits of security, and the field \mathbb{F} is of a 256-bit prime order p .

3 Definition of a decentralized anonymous payment scheme

We introduce the notion of a *decentralized anonymous payment scheme* (DAP scheme), extending the notion of *decentralized e-cash* [MGGR13]. Later, in Section 4, we provide a construction.

3.1 Data structures

We begin by describing, and giving intuition about, the data structures used by a DAP scheme. The algorithms that use and produce these data structures are introduced in Section 3.2.

Basecoin ledger. Our protocol is applied on top of a ledger-based base currency such as Bitcoin; for generality we refer to this base currency as *Basecoin*. At any given time T , all users have access to L_T , the *ledger* at time T , which is a sequence of *transactions*. The ledger is append-only (i.e., $T < T'$ implies that L_T is a prefix of $L_{T'}$).¹¹ The transactions in the ledger include both Basecoin transactions as well as two new transaction types described below.

Public parameters. A list of *public parameters* pp is available to all users in the system. These are generated by a trusted party at the “start of time” and are used by the system’s algorithms.

Addresses. Each user generates at least one *address key pair* $(\text{addr}_{\text{pk}}, \text{addr}_{\text{sk}})$. The public key addr_{pk} is published and enables others to direct payments to the user. The secret key addr_{sk} is used to receive payments sent to addr_{pk} . A user may generate any number of address key pairs.

Coins. A *coin* is a data object \mathbf{c} , to which we associate the following.

- A *coin commitment*, denoted $\text{cm}(\mathbf{c})$: a string that appears on the ledger once \mathbf{c} is *minted*.
- A *coin value*, denoted $v(\mathbf{c})$: the denomination of \mathbf{c} , as measured in basecoins, as an integer between 0 and a maximum value v_{\max} (which is a system parameter).
- A *coin serial number*, denoted $\text{sn}(\mathbf{c})$: a unique string associated with \mathbf{c} , used to prevent double spending.
- A *coin address*, denoted $\text{addr}_{\text{pk}}(\mathbf{c})$: an address public key, representing who owns \mathbf{c} .

Any other quantities associated with a coin \mathbf{c} (e.g., various trapdoors) are implementation details.

New transactions. Besides Basecoin transactions, there are two new types of transactions.

- *Mint transactions.* A mint transaction tx_{Mint} is a tuple $(\text{cm}, v, *)$, where cm is a coin commitment, v is a coin value, and $*$ denotes other (implementation-dependent) information. The transaction tx_{Mint} records that a coin \mathbf{c} with coin commitment cm and value v has been minted.
- *Pour transactions.* A pour transaction tx_{Pour} is a tuple $(\text{rt}, \text{sn}_1^{\text{old}}, \text{sn}_2^{\text{old}}, \text{cm}_1^{\text{new}}, \text{cm}_2^{\text{new}}, v_{\text{pub}}, \text{info}, *)$, where rt is a root of a Merkle tree, $\text{sn}_1^{\text{old}}, \text{sn}_2^{\text{old}}$ are two coin serial numbers, $\text{cm}_1^{\text{new}}, \text{cm}_2^{\text{new}}$ are two coin commitments, v_{pub} is a coin value, info is an arbitrary string, and $*$ denotes other (implementation-dependent) information. The transaction tx_{Pour} records the pouring of two input (and now consumed) coins $\mathbf{c}_1^{\text{old}}, \mathbf{c}_2^{\text{old}}$, with respective serial numbers $\text{sn}_1^{\text{old}}, \text{sn}_2^{\text{old}}$, into two new output coins $\mathbf{c}_1^{\text{new}}, \mathbf{c}_2^{\text{new}}$, with respective coin commitments $\text{cm}_1^{\text{new}}, \text{cm}_2^{\text{new}}$, as well as a public output v_{pub} (which may be zero). Furthermore, tx_{Pour} also records an information string info (perhaps containing information on who is the recipient of v_{pub} basecoins) and that, when this transaction was made, the root of the Merkle tree over coin commitments was rt (see below).

¹¹In reality, the Basecoin ledger (such as the one of Bitcoin) is not perfect and may incur temporary inconsistencies. In this respect our construction is as good as the underlying ledger. We discuss the effects of this on anonymity and mitigations in Section 6.4.

Commitments of minted coins and serial numbers of spent coins. For any given time T ,

- CMList_T denotes the list of all coin commitments appearing in mint and pour transactions in L_T ;
- SNList_T denotes the list of all serial numbers appearing in pour transactions in L_T .

While both of these lists can be deduced from L_T , it will be convenient to think about them as separate (as, in practice, these may be separately maintained for efficiency reasons; cf. Section 8.3).

Merkle tree over commitments. For any given time T , Tree_T denotes a Merkle tree over CMList_T and rt_T its root. Moreover, the function $\text{Path}_T(\text{cm})$ gives the authentication path from a coin commitment cm appearing in CMList_T to the root of Tree_T .¹² For convenience, we assume that L_T also stores $\text{rt}_{T'}$ for all $T' \leq T$ (i.e., it stores all past Merkle tree roots).

3.2 Algorithms

A DAP scheme Π is a tuple of polynomial-time algorithms

(Setup, CreateAddress, Mint, Pour, VerifyTransaction, Receive)

with the following syntax and semantics.

System setup. The algorithm **Setup** generates a list of public parameters:

Setup

- INPUTS: security parameter λ
- OUTPUTS: public parameters pp

The algorithm **Setup** is executed by a trusted party. The resulting public parameters pp are published and made available to all parties (e.g., by embedding them into the protocol’s implementation). The setup is done *only once*; afterwards, no trusted party is needed, and no global secrets or trapdoors are kept.

Creating payment addresses. The algorithm **CreateAddress** generates a new address key pair:

CreateAddress

- INPUTS: public parameters pp
- OUTPUTS: address key pair $(\text{addr}_{\text{pk}}, \text{addr}_{\text{sk}})$

Each user generates at least one address key pair $(\text{addr}_{\text{pk}}, \text{addr}_{\text{sk}})$ in order to receive coins. The public key addr_{pk} is published, while the secret key addr_{sk} is used to redeem coins sent to addr_{pk} . A user may generate any number of address key pairs; doing so does not require any interaction.

Minting coins. The algorithm **Mint** generates a coin (of a given value) and a mint transaction:

Mint

- INPUTS:
 - public parameters pp
 - coin value $v \in \{0, 1, \dots, v_{\max}\}$
 - destination address public key addr_{pk}
- OUTPUTS: coin c and mint transaction tx_{Mint}

A system parameter, v_{\max} , caps the value of any single coin. The output coin c has value v and coin address addr_{pk} ; the output mint transaction tx_{Mint} equals $(\text{cm}, v, *)$, where cm is the coin commitment of c .

¹²While we refer to Merkle trees for simplicity, it is straightforward to extend the definition to allow other data structures representing sets with fast insertion and efficient proofs of membership.

Pouring coins. The Pour algorithm transfers value from input coins into new output coins, marking the input coins as consumed. Moreover, a fraction of the input value may be publicly revealed. Pouring allows users to subdivide coins into smaller denominations, merge coins, and transfer ownership of anonymous coins, or make public payments.¹³

Pour

- INPUTS:
 - public parameters \mathbf{pp}
 - the Merkle root \mathbf{rt}
 - old coins $\mathbf{c}_1^{\text{old}}, \mathbf{c}_2^{\text{old}}$
 - old addresses secret keys $\mathbf{addr}_{\mathbf{sk},1}^{\text{old}}, \mathbf{addr}_{\mathbf{sk},2}^{\text{old}}$
 - authentication path \mathbf{path}_1 from commitment $\mathbf{cm}(\mathbf{c}_1^{\text{old}})$ to root \mathbf{rt} , authentication path \mathbf{path}_2 from commitment $\mathbf{cm}(\mathbf{c}_2^{\text{old}})$ to root \mathbf{rt}
 - new values $v_1^{\text{new}}, v_2^{\text{new}}$
 - new addresses public keys $\mathbf{addr}_{\mathbf{pk},1}^{\text{new}}, \mathbf{addr}_{\mathbf{pk},2}^{\text{new}}$
 - public value v_{pub}
 - transaction string \mathbf{info}
- OUTPUTS: new coins $\mathbf{c}_1^{\text{new}}, \mathbf{c}_2^{\text{new}}$ and pour transaction $\mathbf{tx}_{\text{Pour}}$

Thus, the Pour algorithm takes as input two distinct input coins $\mathbf{c}_1^{\text{old}}, \mathbf{c}_2^{\text{old}}$, along with corresponding address secret keys $\mathbf{addr}_{\mathbf{sk},1}^{\text{old}}, \mathbf{addr}_{\mathbf{sk},2}^{\text{old}}$ (required to redeem the two input coins). To ensure that $\mathbf{c}_1^{\text{old}}, \mathbf{c}_2^{\text{old}}$ have been previously minted, the Pour algorithm also takes as input the Merkle root \mathbf{rt} (allegedly, equal to the root of Merkle tree over all coin commitments so far), along with two authentication paths $\mathbf{path}_1, \mathbf{path}_2$ for the two coin commitments $\mathbf{cm}(\mathbf{c}_1^{\text{old}}), \mathbf{cm}(\mathbf{c}_2^{\text{old}})$. Two input values $v_1^{\text{new}}, v_2^{\text{new}}$ specify the values of two new anonymous coins $\mathbf{c}_1^{\text{new}}, \mathbf{c}_2^{\text{new}}$ to be generated, and two input address public keys $\mathbf{addr}_{\mathbf{pk},1}^{\text{new}}, \mathbf{addr}_{\mathbf{pk},2}^{\text{new}}$ specify the recipients of $\mathbf{c}_1^{\text{new}}, \mathbf{c}_2^{\text{new}}$. A third value, v_{pub} , specifies the amount to be publicly spent (e.g., to redeem coins or pay transaction fees). The sum of output values $v_1^{\text{new}} + v_2^{\text{new}} + v_{\text{pub}}$ must be equal to the sum of the values of the input coins (and cannot exceed v_{max}). Finally, the Pour algorithm also receives an arbitrary string \mathbf{info} , which is bound into the output pour transaction $\mathbf{tx}_{\text{Pour}}$.

The Pour algorithm outputs two new coins $\mathbf{c}_1^{\text{new}}, \mathbf{c}_2^{\text{new}}$ and a pour transaction $\mathbf{tx}_{\text{Pour}}$. The transaction $\mathbf{tx}_{\text{Pour}}$ equals $(\mathbf{rt}, \mathbf{sn}_1^{\text{old}}, \mathbf{sn}_2^{\text{old}}, \mathbf{cm}_1^{\text{new}}, \mathbf{cm}_2^{\text{new}}, v_{\text{pub}}, \mathbf{info}, *)$, where $\mathbf{cm}_1^{\text{new}}, \mathbf{cm}_2^{\text{new}}$ are the two coin commitments of the two output coins, and $*$ denotes other (implementation-dependent) information. Crucially, $\mathbf{tx}_{\text{Pour}}$ reveals only one value, the public value v_{pub} (which may be zero); it does not reveal the payment addresses or values of the old or new coins.

Verifying transactions. The algorithm VerifyTransaction checks the validity of a transaction:

VerifyTransaction

- INPUTS:
 - public parameters \mathbf{pp}
 - a (mint or pour) transaction \mathbf{tx}
 - the current ledger L
- OUTPUTS: bit b , equals 1 iff the transaction is valid

Both mint and pour transactions must be verified before being considered well-formed. In practice, transactions can be verified by the nodes in the distributed system maintaining the ledger, as well

¹³We consider pours with 2 inputs and 2 outputs, for simplicity and (as discussed in Section 1.3) without loss of generality.

as by users who rely on these transactions.

Receiving coins. The algorithm `Receive` scans the ledger and retrieves unspent coins paid to a particular user address:

`Receive`

- INPUTS:
 - recipient address key pair (addr_{pk} , addr_{sk})
 - the current ledger L
- OUTPUTS: set of (unspent) received coins

When a user with address key pair (addr_{pk} , addr_{sk}) wishes to receive payments sent to addr_{pk} , he uses the `Receive` algorithm to scan the ledger. For each payment to addr_{pk} appearing in the ledger, `Receive` outputs the corresponding coins whose serial numbers do not appear on the ledger L . Coins received in this way may be spent, just like minted coins, using the `Pour` algorithm. (We only require `Receive` to detect coins paid to addr_{pk} via the `Pour` algorithm and not also detect coins minted by the user himself.)

Next, we describe completeness (Section 3.3) and security (Section 3.4).

3.3 Completeness

Completeness of a DAP scheme requires that unspent coins can be spent. More precisely, consider a *ledger sampler* \mathcal{S} outputting a ledger L . If \mathbf{c}_1 and \mathbf{c}_2 are two coins whose coin commitments appear in (valid) transactions on L , but their serial numbers do not appear in L , then \mathbf{c}_1 and \mathbf{c}_2 can be spent using `Pour`. Namely, running `Pour` results in a pour transaction tx_{Pour} that `VerifyTransaction` accepts, and the new coins can be received by the intended recipients (by using `Receive`); moreover, tx_{Pour} correctly records the intended v_{pub} and transaction string info . This property is formalized via an *incompleteness experiment* `INCOMP`.

Definition 3.1. A DAP scheme $\Pi = (\text{Setup}, \text{CreateAddress}, \text{Mint}, \text{Pour}, \text{VerifyTransaction}, \text{Receive})$ is **complete** if no polynomial-size ledger sampler \mathcal{S} wins `INCOMP` with more than negligible probability. (See Appendix B for details.)

3.4 Security

Security of a DAP scheme is characterized by three properties, which we call *ledger indistinguishability*, *transaction non-malleability*, and *balance*.

Definition 3.2. A DAP scheme $\Pi = (\text{Setup}, \text{CreateAddress}, \text{Mint}, \text{Pour}, \text{VerifyTransaction}, \text{Receive})$ is **secure** if it satisfies ledger indistinguishability, transaction non-malleability, and balance.

Below, we provide an informal overview of each property, and defer formal definitions to Appendix C.

Each property is formalized as a game between an adversary \mathcal{A} and a challenger \mathcal{C} . In each game, the behavior of honest parties is realized via a DAP scheme oracle \mathcal{O}^{DAP} , which maintains a ledger L and provides an interface for executing `CreateAddress`, `Mint`, `Pour` and `Receive` algorithms for honest parties. To elicit behavior from honest parties, \mathcal{A} passes a query to \mathcal{C} , which (after sanity checks) proxies the query to \mathcal{O}^{DAP} . For each query that requests an honest party to perform an action, \mathcal{A} specifies identities of previous transactions and the input values, and learns the resulting transaction, but not any of the secrets or trapdoors involved in producing that transaction. The oracle \mathcal{O}^{DAP} also provides an `Insert` query that allows \mathcal{A} to directly add arbitrary transactions to the ledger L .

Ledger indistinguishability. This property captures the requirement that the ledger reveals no new information to the adversary beyond the publicly-revealed information (values of minted coins, public values, information strings, total number of transactions, etc.), even when the adversary can adaptively induce honest parties to perform DAP operations of his choice. That is, no bounded adversary \mathcal{A} can distinguish between two ledgers L_0 and L_1 , constructed by \mathcal{A} using queries to two DAP scheme oracles, when the queries to the two oracles are *publicly consistent*: they have matching type and are identical in terms of publicly-revealed information and the information related to addresses controlled by \mathcal{A} .

Ledger indistinguishability is formalized by an experiment L-IND that proceeds as follows. First, a challenger samples a random bit b and initializes two DAP scheme oracles $\mathcal{O}_0^{\text{DAP}}$ and $\mathcal{O}_1^{\text{DAP}}$, maintaining ledgers L_0 and L_1 . Throughout, the challenger allows \mathcal{A} to issue queries to $\mathcal{O}_0^{\text{DAP}}$ and $\mathcal{O}_1^{\text{DAP}}$, thus controlling the behavior of honest parties on L_0 and L_1 . The challenger provides the adversary with the view of both ledgers, but in randomized order: $L_{\text{Left}} := L_b$ and $L_{\text{Right}} := L_{1-b}$. The adversary’s goal is to distinguish whether the view he sees corresponds to $(L_{\text{Left}}, L_{\text{Right}}) = (L_0, L_1)$, i.e. $b = 0$, or to $(L_{\text{Left}}, L_{\text{Right}}) = (L_1, L_0)$, i.e. $b = 1$.

At each round of the experiment, the adversary issues queries in pairs Q, Q' of matching query type. If the query type is **CreateAddress**, then the same address is generated at both oracles. If it is to **Mint**, **Pour** or **Receive**, then Q is forwarded to L_0 and Q' to L_1 ; for **Insert** queries, query Q is forwarded to L_{Left} and Q' is forwarded to L_{Right} . The adversary’s queries are restricted in the sense that they must maintain the *public consistency* of the two ledgers. For example, the public values for **Pour** queries must be the same, as well as minted amounts for **Mint** queries.

At the conclusion of the experiment, \mathcal{A} outputs a guess b' , and wins when $b = b'$. Ledger indistinguishability requires that \mathcal{A} wins L-IND with probability at most negligibly greater than $1/2$.

Transaction non-malleability. This property requires that no bounded adversary \mathcal{A} can alter any of the data stored within a (valid) pour transaction tx_{Pour} . This *transaction non-malleability* prevents malicious attackers from modifying others’ transactions before they are added to the ledger (e.g., by re-targeting the Basecoin public output of a pour transaction).

Transaction non-malleability is formalized by an experiment TR-NM , in which \mathcal{A} adaptively interacts with a DAP scheme oracle \mathcal{O}^{DAP} and then outputs a pour transaction tx^* . Letting \mathcal{T} denote the set of pour transactions returned by \mathcal{O}^{DAP} , and L denote the final ledger, \mathcal{A} wins the game if there exists $\text{tx} \in \mathcal{T}$, such that (i) $\text{tx}^* \neq \text{tx}$; (ii) tx^* reveals a serial number contained in tx ; and (iii) both tx and tx^* are valid with respect to the ledger L' containing all transactions preceding tx on L . In other words, \mathcal{A} wins the game if tx^* manages to modify some previous pour transaction to spend the same coin in a different way.

Transaction non-malleability requires that \mathcal{A} wins TR-NM with only negligible probability. (Note that \mathcal{A} can of course produce valid pour transactions that are unrelated to those in \mathcal{T} ; the condition that tx^* reveals a serial number of a previously-spent coin captures non-malleability.)

Balance. This property requires that no bounded adversary \mathcal{A} can own more money than what he minted or received via payments from others.

Balance is formalized by an experiment BAL , in which \mathcal{A} adaptively interacts with a DAP scheme oracle \mathcal{O}^{DAP} and then outputs a set of coins S_{coin} . Letting ADDR be set of addresses returned by **CreateAddress** queries (i.e., addresses of “honest” users), \mathcal{A} wins the game if the total value he can spend or has spent (either as coins or Basecoin public outputs) is greater than the value he has minted or received. That is, \mathcal{A} wins if $v_{\text{Unspent}} + v_{\text{Basecoin}} + v_{\mathcal{A} \rightarrow \text{ADDR}} > v_{\text{Mint}} + v_{\text{ADDR} \rightarrow \mathcal{A}}$ where: (i) v_{Unspent} is the total value of unspent coins in S_{coin} ; (ii) v_{Basecoin} is the total value of public outputs placed by \mathcal{A} on the ledger; (iii) v_{Mint} is the total value of \mathcal{A} ’s mint transactions; (iv) $v_{\text{ADDR} \rightarrow \mathcal{A}}$ is the total value of payments received by \mathcal{A} from addresses in ADDR ; (v) $v_{\mathcal{A} \rightarrow \text{ADDR}}$ is the total value

of payments sent by \mathcal{A} to addresses in ADDR .

Balance requires that \mathcal{A} wins BAL with only negligible probability.

4 Construction of a decentralized anonymous payment scheme

We show how to construct a DAP scheme (introduced in Section 3) using zk-SNARKs and other building blocks. Later, in Section 5, we give a concrete instantiation of this construction.

4.1 Cryptographic building blocks

We first introduce notation for the standard cryptographic building blocks that we use. We assume familiarity with the definitions of these building blocks; for more details, see, e.g., [KL07]. Throughout, λ denotes the security parameter.

Collision-resistant hashing. We use a collision-resistant hash function $\text{CRH}: \{0, 1\}^* \rightarrow \{0, 1\}^{O(\lambda)}$.

Pseudorandom functions. We use a pseudorandom function family $\text{PRF} = \{\text{PRF}_x: \{0, 1\}^* \rightarrow \{0, 1\}^{O(\lambda)}\}_x$ where x denotes the seed. From PRF_x , we derive three “non-overlapping” pseudorandom functions, chosen arbitrarily as $\text{PRF}_x^{\text{addr}}(z) := \text{PRF}_x(00\|z)$, $\text{PRF}_x^{\text{sn}}(z) := \text{PRF}_x(01\|z)$, $\text{PRF}_x^{\text{pk}}(z) := \text{PRF}_x(10\|z)$. Furthermore, we assume that PRF^{sn} is also collision resistant, in the sense that it is infeasible to find $(x, z) \neq (x', z')$ such that $\text{PRF}_x^{\text{sn}}(z) = \text{PRF}_{x'}^{\text{sn}}(z')$.

Statistically-hiding commitments. We use a commitment scheme COMM where the binding property holds computationally, while the hiding property holds statistically. It is denoted $\{\text{COMM}_x: \{0, 1\}^* \rightarrow \{0, 1\}^{O(\lambda)}\}_x$ where x denotes the commitment trapdoor. Namely, to reveal a commitment cm to a value z , it suffices to provide z and the trapdoor x ; then one can check that $\text{cm} = \text{COMM}_x(z)$.

One-time strongly-unforgeable digital signatures. We use a digital signature scheme $\text{Sig} = (\mathcal{G}_{\text{sig}}, \mathcal{K}_{\text{sig}}, \mathcal{S}_{\text{sig}}, \mathcal{V}_{\text{sig}})$ that works as follows.

- $\mathcal{G}_{\text{sig}}(1^\lambda) \rightarrow \text{pp}_{\text{sig}}$. Given a security parameter λ (presented in unary), \mathcal{G}_{sig} samples public parameters pp_{sig} for the encryption scheme.
- $\mathcal{K}_{\text{sig}}(\text{pp}_{\text{sig}}) \rightarrow (\text{pk}_{\text{sig}}, \text{sk}_{\text{sig}})$. Given public parameters pp_{sig} , \mathcal{K}_{sig} samples a public key and a secret key for a single user.
- $\mathcal{S}_{\text{sig}}(\text{sk}_{\text{sig}}, m) \rightarrow \sigma$. Given a secret key sk_{sig} and a message m , \mathcal{S}_{sig} signs m to obtain a signature σ .
- $\mathcal{V}_{\text{sig}}(\text{pk}_{\text{sig}}, m, \sigma) \rightarrow b$. Given a public key pk_{sig} , message m , and signature σ , \mathcal{V}_{sig} outputs $b = 1$ if the signature σ is valid for message m ; else it outputs $b = 0$.

The signature scheme Sig satisfies the security property of *one-time strong unforgeability against chosen-message attacks* (SUF-1CMA security).

Key-private public-key encryption. We use a public-key encryption scheme $\text{Enc} = (\mathcal{G}_{\text{enc}}, \mathcal{K}_{\text{enc}}, \mathcal{E}_{\text{enc}}, \mathcal{D}_{\text{enc}})$ that works as follows.

- $\mathcal{G}_{\text{enc}}(1^\lambda) \rightarrow \text{pp}_{\text{enc}}$. Given a security parameter λ (presented in unary), \mathcal{G}_{enc} samples public parameters pp_{enc} for the encryption scheme.
- $\mathcal{K}_{\text{enc}}(\text{pp}_{\text{enc}}) \rightarrow (\text{pk}_{\text{enc}}, \text{sk}_{\text{enc}})$. Given public parameters pp_{enc} , \mathcal{K}_{enc} samples a public key and a secret key for a single user.
- $\mathcal{E}_{\text{enc}}(\text{pk}_{\text{enc}}, m) \rightarrow c$. Given a public key pk_{enc} and a message m , \mathcal{E}_{enc} encrypts m to obtain a ciphertext c .
- $\mathcal{D}_{\text{enc}}(\text{sk}_{\text{enc}}, c) \rightarrow m$. Given a secret key sk_{enc} and a ciphertext c , \mathcal{D}_{enc} decrypts c to produce a message m (or \perp if decryption fails).

The encryption scheme Enc satisfies two security properties: (i) *ciphertext indistinguishability under chosen-ciphertext attack* (IND-CCA security); and (ii) *key indistinguishability under chosen-ciphertext*

attack (IK-CCA security). While the first property is standard, the second is less known; informally, IK-CCA requires that ciphertexts cannot be linked to the public key used to encrypt them, or to other ciphertexts encrypted with the same public key. For definitions, we refer the reader to [BBDP01].

4.2 zk-SNARKs for pouring coins

As outlined in Section 1.3, our construction invokes a zk-SNARK for a specific NP statement, POUR, which we now define. We first recall the context motivating POUR. When a user u *pours* “old” coins $\mathbf{c}_1^{\text{old}}, \mathbf{c}_2^{\text{old}}$ into new coins $\mathbf{c}_1^{\text{new}}, \mathbf{c}_2^{\text{new}}$, a corresponding pour transaction

$$\mathbf{tx}_{\text{Pour}} = (\mathbf{rt}, \mathbf{sn}_1^{\text{old}}, \mathbf{sn}_2^{\text{old}}, \mathbf{cm}_1^{\text{new}}, \mathbf{cm}_2^{\text{new}}, v_{\text{pub}}, \mathbf{info}, *)$$

is generated. In our construction, we need to provide evidence in “ $*$ ” that various conditions were respected by the pour operation. Concretely, $\mathbf{tx}_{\text{Pour}}$ should demonstrate that (i) u owns $\mathbf{c}_1^{\text{old}}, \mathbf{c}_2^{\text{old}}$; (ii) coin commitments for $\mathbf{c}_1^{\text{old}}, \mathbf{c}_2^{\text{old}}$ appear somewhere on the ledger; (iii) the revealed serial numbers $\mathbf{sn}_1^{\text{old}}, \mathbf{sn}_2^{\text{old}}$ are of $\mathbf{c}_1^{\text{old}}, \mathbf{c}_2^{\text{old}}$; (iv) the revealed coin commitments $\mathbf{cm}_1^{\text{new}}, \mathbf{cm}_2^{\text{new}}$ are of $\mathbf{c}_1^{\text{new}}, \mathbf{c}_2^{\text{new}}$; (v) balance is preserved. Our construction achieves this by including a zk-SNARK proof π_{POUR} for the statement POUR which checks the above invariants (as well as others needed for non-malleability).

The statement POUR. Concretely, the NP statement POUR is defined as follows.

- Instances are of the form $x = (\mathbf{rt}, \mathbf{sn}_1^{\text{old}}, \mathbf{sn}_2^{\text{old}}, \mathbf{cm}_1^{\text{new}}, \mathbf{cm}_2^{\text{new}}, v_{\text{pub}}, h_{\text{Sig}}, h_1, h_2)$. Thus, an instance x specifies a root \mathbf{rt} for a CRH-based Merkle tree (over the list of commitments so far), the two serial numbers of the consumed coins, two coin commitments for the two new coins, a public value, and fields h_{Sig}, h_1, h_2 used for non-malleability.
- Witnesses are of the form $a = (\mathbf{path}_1, \mathbf{path}_2, \mathbf{c}_1^{\text{old}}, \mathbf{c}_2^{\text{old}}, \mathbf{addr}_{\text{pk},1}^{\text{old}}, \mathbf{addr}_{\text{pk},2}^{\text{old}}, \mathbf{c}_1^{\text{new}}, \mathbf{c}_2^{\text{new}})$ where, for each $i \in \{1, 2\}$:

$$\begin{aligned} \mathbf{c}_i^{\text{old}} &= (\mathbf{addr}_{\text{pk},i}^{\text{old}}, v_i^{\text{old}}, \rho_i^{\text{old}}, r_i^{\text{old}}, s_i^{\text{old}}, \mathbf{cm}_i^{\text{old}}) , \\ \mathbf{c}_i^{\text{new}} &= (\mathbf{addr}_{\text{pk},i}^{\text{new}}, v_i^{\text{new}}, \rho_i^{\text{new}}, r_i^{\text{new}}, s_i^{\text{new}}, \mathbf{cm}_i^{\text{new}}) \text{ for the same } \mathbf{cm}_i^{\text{new}} \text{ as in } x, \\ \mathbf{addr}_{\text{pk},i}^{\text{old}} &= (a_{\text{pk},i}^{\text{old}}, \mathbf{pk}_{\text{enc},i}^{\text{old}}) , \\ \mathbf{addr}_{\text{pk},i}^{\text{new}} &= (a_{\text{pk},i}^{\text{new}}, \mathbf{pk}_{\text{enc},i}^{\text{new}}) , \\ \mathbf{addr}_{\text{sk},i}^{\text{old}} &= (a_{\text{sk},i}^{\text{old}}, \mathbf{sk}_{\text{enc},i}^{\text{old}}) . \end{aligned}$$

Thus, a witness a specifies authentication paths for the two new coin commitments, the entirety of coin information about both the old and new coins, and address secret keys for the old coins.

Given a POUR instance x , a witness a is valid for x if the following holds:

1. For each $i \in \{1, 2\}$:
 - (a) The coin commitment $\mathbf{cm}_i^{\text{old}}$ of $\mathbf{c}_i^{\text{old}}$ appears on the ledger, i.e., \mathbf{path}_i is a valid authentication path for leaf $\mathbf{cm}_i^{\text{old}}$ with respect to root \mathbf{rt} , in a CRH-based Merkle tree.
 - (b) The address secret key $a_{\text{sk},i}^{\text{old}}$ matches the address public key of $\mathbf{c}_i^{\text{old}}$, i.e., $a_{\text{pk},i}^{\text{old}} = \text{PRF}_{a_{\text{sk},i}^{\text{old}}}^{\text{addr}}(0)$.
 - (c) The serial number $\mathbf{sn}_i^{\text{old}}$ of $\mathbf{c}_i^{\text{old}}$ is computed correctly, i.e., $\mathbf{sn}_i^{\text{old}} = \text{PRF}_{a_{\text{sk},i}^{\text{old}}}^{\text{sn}}(\rho_i^{\text{old}})$.
 - (d) The coin $\mathbf{c}_i^{\text{old}}$ is well-formed, i.e., $\mathbf{cm}_i^{\text{old}} = \text{COMM}_{s_i^{\text{old}}}(\text{COMM}_{r_i^{\text{old}}}(a_{\text{pk},i}^{\text{old}} \parallel \rho_i^{\text{old}}) \parallel v_i^{\text{old}})$.
 - (e) The coin $\mathbf{c}_i^{\text{new}}$ is well-formed, i.e., $\mathbf{cm}_i^{\text{new}} = \text{COMM}_{s_i^{\text{new}}}(\text{COMM}_{r_i^{\text{new}}}(a_{\text{pk},i}^{\text{new}} \parallel \rho_i^{\text{new}}) \parallel v_i^{\text{new}})$.
 - (f) The address secret key $a_{\text{sk},i}^{\text{old}}$ ties h_{Sig} to h_i , i.e., $h_i = \text{PRF}_{a_{\text{sk},i}^{\text{old}}}^{\text{pk}}(i \parallel h_{\text{Sig}})$.

2. Balance is preserved: $v_1^{\text{new}} + v_2^{\text{new}} + v_{\text{pub}} = v_1^{\text{old}} + v_2^{\text{old}}$ (with $v_1^{\text{old}}, v_2^{\text{old}} \geq 0$ and $v_1^{\text{old}} + v_2^{\text{old}} \leq v_{\max}$).

Recall that in this paper zk-SNARKs are relative to the language of arithmetic circuit satisfiability (see Section 2); thus, we express the checks in POUR via an arithmetic circuit, denoted C_{POUR} . In particular, the depth d_{tree} of the Merkle tree needs to be hardcoded in C_{POUR} , and we thus make it a parameter of our construction (see below); the maximum number of supported coins is then $2^{d_{\text{tree}}}$.

4.3 Algorithm constructions

We proceed to describe the construction of the DAP scheme $\Pi = (\text{Setup}, \text{CreateAddress}, \text{Mint}, \text{Pour}, \text{VerifyTransaction}, \text{Receive})$ whose intuition was given in Section 1.3. Figure 2 gives the pseudocode for each one of the six algorithms in Π , in terms of the building blocks introduced in Section 4.1 and Section 4.2. In the construction, we hardcode two quantities: the maximum value of a coin, v_{\max} , and the depth of the Merkle tree, d_{tree} .

4.4 Completeness and security

Our main theorem states that the above construction is indeed a DAP scheme.

Theorem 4.1. The tuple $\Pi = (\text{Setup}, \text{CreateAddress}, \text{Mint}, \text{Pour}, \text{VerifyTransaction}, \text{Receive})$, as defined in Section 4.3, is a complete (cf. Definition 3.1) and secure (cf. Definition 3.2) DAP scheme.

We provide a proof of Theorem 4.1 in Appendix D. We note that our construction can be modified to yield statistical (i.e., everlasting) anonymity; see the discussion in Section 8.1.

Remark (trusted setup). Security of Π relies on a trusted party running `Setup` to generate the public parameters (once and for all). This trust is needed for the transaction non-malleability and balance properties but not for ledger indistinguishability. Thus, even if a powerful espionage agency were to corrupt the setup, anonymity will *still* be maintained. Moreover, if one wishes to mitigate the trust requirements of this step, one can conduct the computation of `Setup` using secure multiparty computation techniques; we leave this to future work.

Remark (use of `pp`). According to the definition of a DAP scheme (see Section 3), the public parameters `pp` are given as input to each one of the six algorithms; this is also how we presented our construction in Figure 2. However, in our construction, the public parameters `pp` equal a tuple $(\mathbf{pk}_{\text{POUR}}, \mathbf{vk}_{\text{POUR}}, \mathbf{pp}_{\text{enc}}, \mathbf{pp}_{\text{sig}})$, and not every algorithm needs every component of `pp`. Concretely, `CreateAddress` only needs `ppenc`; `Mint` only the security parameter λ ; `Pour` only $\mathbf{pk}_{\text{POUR}}$ and \mathbf{pp}_{sig} ; `VerifyTransaction` only $\mathbf{vk}_{\text{POUR}}$; and `Receive` only λ . In particular, since we rely on zk-SNARKs to prove/verify POUR, $\mathbf{pk}_{\text{POUR}}$ is of constant, but large, size, and is only required by `Pour`. All other components of `pp` are of small constant size.

Remark (checking received coins in ledger). The algorithm `Receive` tests whether the serial number of a received coin already appears on the ledger, in order not to output coins that the user has already received and spent by himself. Other users are, in any case, unable to spend coins addressed to this user.

5 Zerocash

We describe a concrete instantiation of a DAP scheme; this instantiation forms the basis of Zerocash. Later, in Section 6, we discuss how Zerocash can be integrated with existing ledger-based currencies.

<p>Setup</p> <ul style="list-style-type: none"> • INPUTS: security parameter λ • OUTPUTS: public parameters pp <ol style="list-style-type: none"> 1. Construct C_{POUR} for POUR at security λ. 2. Compute $(\mathbf{pk}_{\text{POUR}}, \mathbf{vk}_{\text{POUR}}) := \text{KeyGen}(1^\lambda, C_{\text{POUR}})$. 3. Compute $\mathbf{pp}_{\text{enc}} := \mathcal{G}_{\text{enc}}(1^\lambda)$. 4. Compute $\mathbf{pp}_{\text{sig}} := \mathcal{G}_{\text{sig}}(1^\lambda)$. 5. Set $\mathbf{pp} := (\mathbf{pk}_{\text{POUR}}, \mathbf{vk}_{\text{POUR}}, \mathbf{pp}_{\text{enc}}, \mathbf{pp}_{\text{sig}})$. 6. Output pp. <p>CreateAddress</p> <ul style="list-style-type: none"> • INPUTS: public parameters pp • OUTPUTS: address key pair $(\mathbf{addr}_{\text{pk}}, \mathbf{addr}_{\text{sk}})$ <ol style="list-style-type: none"> 1. Compute $(\mathbf{pk}_{\text{enc}}, \mathbf{sk}_{\text{enc}}) := \mathcal{K}_{\text{enc}}(\mathbf{pp}_{\text{enc}})$. 2. Randomly sample a PRF^{addr} seed a_{sk}. 3. Compute $a_{\text{pk}} = \text{PRF}_{a_{\text{sk}}}^{\text{addr}}(0)$. 4. Set $\mathbf{addr}_{\text{pk}} := (a_{\text{pk}}, \mathbf{pk}_{\text{enc}})$. 5. Set $\mathbf{addr}_{\text{sk}} := (a_{\text{sk}}, \mathbf{sk}_{\text{enc}})$. 6. Output $(\mathbf{addr}_{\text{pk}}, \mathbf{addr}_{\text{sk}})$. <p>Mint</p> <ul style="list-style-type: none"> • INPUTS: <ul style="list-style-type: none"> – public parameters pp – coin value $v \in \{0, 1, \dots, v_{\text{max}}\}$ – destination address public key $\mathbf{addr}_{\text{pk}}$ • OUTPUTS: coin \mathbf{c} and mint transaction $\mathbf{tx}_{\text{Mint}}$ <ol style="list-style-type: none"> 1. Parse $\mathbf{addr}_{\text{pk}}$ as $(a_{\text{pk}}, \mathbf{pk}_{\text{enc}})$. 2. Randomly sample a PRF^{sn} seed ρ. 3. Randomly sample two COMM trapdoors r, s. 4. Compute $k := \text{COMM}_r(a_{\text{pk}} \parallel \rho)$. 5. Compute $\mathbf{cm} := \text{COMM}_s(v \parallel k)$. 6. Set $\mathbf{c} := (\mathbf{addr}_{\text{pk}}, v, \rho, r, s, \mathbf{cm})$. 7. Set $\mathbf{tx}_{\text{Mint}} := (\mathbf{cm}, v, *)$, where $* := (k, s)$. 8. Output \mathbf{c} and $\mathbf{tx}_{\text{Mint}}$. <p>VerifyTransaction</p> <ul style="list-style-type: none"> • INPUTS: <ul style="list-style-type: none"> – public parameters pp – a (mint or pour) transaction tx – the current ledger L • OUTPUTS: bit b, equals 1 iff the transaction is valid <ol style="list-style-type: none"> 1. If given a mint transaction $\mathbf{tx} = \mathbf{tx}_{\text{Mint}}$: <ol style="list-style-type: none"> (a) Parse $\mathbf{tx}_{\text{Mint}}$ as $(\mathbf{cm}, v, *)$, and $*$ as (k, s). (b) Set $\mathbf{cm}' := \text{COMM}_s(v \parallel k)$. (c) Output $b := 1$ if $\mathbf{cm} = \mathbf{cm}'$, else output $b := 0$. 2. If given a pour transaction $\mathbf{tx} = \mathbf{tx}_{\text{Pour}}$: <ol style="list-style-type: none"> (a) Parse $\mathbf{tx}_{\text{Pour}}$ as $(\mathbf{rt}, \mathbf{sn}_1^{\text{old}}, \mathbf{sn}_2^{\text{old}}, \mathbf{cm}_1^{\text{new}}, \mathbf{cm}_2^{\text{new}}, v_{\text{pub}}, \mathbf{info}, *)$, and $*$ as $(\mathbf{pk}_{\text{sig}}, h_1, h_2, \pi_{\text{POUR}}, \mathbf{C}_1, \mathbf{C}_2, \sigma)$. (b) If $\mathbf{sn}_1^{\text{old}}$ or $\mathbf{sn}_2^{\text{old}}$ appears on L (or $\mathbf{sn}_1^{\text{old}} = \mathbf{sn}_2^{\text{old}}$), output $b := 0$. (c) If the Merkle root \mathbf{rt} does not appear on L, output $b := 0$. (d) Compute $h_{\text{Sig}} := \text{CRH}(\mathbf{pk}_{\text{sig}})$. (e) Set $x := (\mathbf{rt}, \mathbf{sn}_1^{\text{old}}, \mathbf{sn}_2^{\text{old}}, \mathbf{cm}_1^{\text{new}}, \mathbf{cm}_2^{\text{new}}, v_{\text{pub}}, h_{\text{Sig}}, h_1, h_2)$. (f) Set $m := (x, \pi_{\text{POUR}}, \mathbf{info}, \mathbf{C}_1, \mathbf{C}_2)$. (g) Compute $b := \mathcal{V}_{\text{sig}}(\mathbf{pk}_{\text{sig}}, m, \sigma)$. (h) Compute $b' := \text{Verify}(\mathbf{vk}_{\text{POUR}}, x, \pi_{\text{POUR}})$, and output $b \wedge b'$. 	<p>Pour</p> <ul style="list-style-type: none"> • INPUTS: <ul style="list-style-type: none"> – public parameters pp – the Merkle root \mathbf{rt} – old coins $\mathbf{c}_1^{\text{old}}, \mathbf{c}_2^{\text{old}}$ – old addresses secret keys $\mathbf{addr}_{\text{sk},1}^{\text{old}}, \mathbf{addr}_{\text{sk},2}^{\text{old}}$ – path \mathbf{path}_1 from commitment $\text{cm}(\mathbf{c}_1^{\text{old}})$ to root \mathbf{rt}, path \mathbf{path}_2 from commitment $\text{cm}(\mathbf{c}_2^{\text{old}})$ to root \mathbf{rt} – new values $v_1^{\text{new}}, v_2^{\text{new}}$ – new addresses public keys $\mathbf{addr}_{\text{pk},1}^{\text{new}}, \mathbf{addr}_{\text{pk},2}^{\text{new}}$ – public value v_{pub} – transaction string info • OUTPUTS: new coins $\mathbf{c}_1^{\text{new}}, \mathbf{c}_2^{\text{new}}$ and pour transaction $\mathbf{tx}_{\text{Pour}}$ <ol style="list-style-type: none"> 1. For each $i \in \{1, 2\}$: <ol style="list-style-type: none"> (a) Parse $\mathbf{c}_i^{\text{old}}$ as $(\mathbf{addr}_{\text{pk},i}^{\text{old}}, v_i^{\text{old}}, \rho_i^{\text{old}}, r_i^{\text{old}}, s_i^{\text{old}}, \mathbf{cm}_i^{\text{old}})$. (b) Parse $\mathbf{addr}_{\text{sk},i}^{\text{old}}$ as $(a_{\text{sk},i}^{\text{old}}, \mathbf{sk}_{\text{enc},i}^{\text{old}})$. (c) Compute $\mathbf{sn}_i^{\text{old}} := \text{PRF}_{a_{\text{sk},i}^{\text{old}}}^{\text{sn}}(\rho_i^{\text{old}})$. (d) Parse $\mathbf{addr}_{\text{pk},i}^{\text{new}}$ as $(a_{\text{pk},i}^{\text{new}}, \mathbf{pk}_{\text{enc},i}^{\text{new}})$. (e) Randomly sample a PRF^{sn} seed ρ_i^{new}. (f) Randomly sample two COMM trapdoors $r_i^{\text{new}}, s_i^{\text{new}}$. (g) Compute $k_i^{\text{new}} := \text{COMM}_{r_i^{\text{new}}}^{\text{sn}}(a_{\text{pk},i}^{\text{new}} \parallel \rho_i^{\text{new}})$. (h) Compute $\mathbf{cm}_i^{\text{new}} := \text{COMM}_{s_i^{\text{new}}}^{\text{sn}}(v_i^{\text{new}} \parallel k_i^{\text{new}})$. (i) Set $\mathbf{c}_i^{\text{new}} := (\mathbf{addr}_{\text{pk},i}^{\text{new}}, v_i^{\text{new}}, \rho_i^{\text{new}}, r_i^{\text{new}}, s_i^{\text{new}}, \mathbf{cm}_i^{\text{new}})$. (j) Set $\mathbf{C}_i := \mathcal{E}_{\text{enc}}(\mathbf{pk}_{\text{enc},i}^{\text{new}}, (v_i^{\text{new}}, \rho_i^{\text{new}}, r_i^{\text{new}}, s_i^{\text{new}}))$. 2. Generate $(\mathbf{pk}_{\text{sig}}, \mathbf{sk}_{\text{sig}}) := \mathcal{K}_{\text{sig}}(\mathbf{pp}_{\text{sig}})$. 3. Compute $h_{\text{Sig}} := \text{CRH}(\mathbf{pk}_{\text{sig}})$. 4. Compute $h_1 := \text{PRF}_{a_{\text{sk},1}^{\text{old}}}^{\text{pk}}(1 \parallel h_{\text{Sig}})$ and $h_2 := \text{PRF}_{a_{\text{sk},2}^{\text{old}}}^{\text{pk}}(2 \parallel h_{\text{Sig}})$. 5. Set $x := (\mathbf{rt}, \mathbf{sn}_1^{\text{old}}, \mathbf{sn}_2^{\text{old}}, \mathbf{cm}_1^{\text{new}}, \mathbf{cm}_2^{\text{new}}, v_{\text{pub}}, h_{\text{Sig}}, h_1, h_2)$. 6. Set $a := (\mathbf{path}_1, \mathbf{path}_2, \mathbf{c}_1^{\text{old}}, \mathbf{c}_2^{\text{old}}, \mathbf{addr}_{\text{sk},1}^{\text{old}}, \mathbf{addr}_{\text{sk},2}^{\text{old}}, \mathbf{c}_1^{\text{new}}, \mathbf{c}_2^{\text{new}})$. 7. Compute $\pi_{\text{POUR}} := \text{Prove}(\mathbf{pk}_{\text{POUR}}, x, a)$. 8. Set $m := (x, \pi_{\text{POUR}}, \mathbf{info}, \mathbf{C}_1, \mathbf{C}_2)$. 9. Compute $\sigma := \mathcal{S}_{\text{sig}}(\mathbf{sk}_{\text{sig}}, m)$. 10. Set $\mathbf{tx}_{\text{Pour}} := (\mathbf{rt}, \mathbf{sn}_1^{\text{old}}, \mathbf{sn}_2^{\text{old}}, \mathbf{cm}_1^{\text{new}}, \mathbf{cm}_2^{\text{new}}, v_{\text{pub}}, \mathbf{info}, *)$, where $* := (\mathbf{pk}_{\text{sig}}, h_1, h_2, \pi_{\text{POUR}}, \mathbf{C}_1, \mathbf{C}_2, \sigma)$. 11. Output $\mathbf{c}_1^{\text{new}}, \mathbf{c}_2^{\text{new}}$ and $\mathbf{tx}_{\text{Pour}}$. <p>Receive</p> <ul style="list-style-type: none"> • INPUTS: <ul style="list-style-type: none"> – public parameters pp – recipient address key pair $(\mathbf{addr}_{\text{pk}}, \mathbf{addr}_{\text{sk}})$ – the current ledger L • OUTPUTS: set of received coins <ol style="list-style-type: none"> 1. Parse $\mathbf{addr}_{\text{pk}}$ as $(a_{\text{pk}}, \mathbf{pk}_{\text{enc}})$. 2. Parse $\mathbf{addr}_{\text{sk}}$ as $(a_{\text{sk}}, \mathbf{sk}_{\text{enc}})$. 3. For each Pour transaction $\mathbf{tx}_{\text{Pour}}$ on the ledger: <ol style="list-style-type: none"> (a) Parse $\mathbf{tx}_{\text{Pour}}$ as $(\mathbf{rt}, \mathbf{sn}_1^{\text{old}}, \mathbf{sn}_2^{\text{old}}, \mathbf{cm}_1^{\text{new}}, \mathbf{cm}_2^{\text{new}}, v_{\text{pub}}, \mathbf{info}, *)$, and $*$ as $(\mathbf{pk}_{\text{sig}}, h_1, h_2, \pi_{\text{POUR}}, \mathbf{C}_1, \mathbf{C}_2, \sigma)$. (b) For each $i \in \{1, 2\}$: <ol style="list-style-type: none"> i. Compute $(v_i, \rho_i, r_i, s_i) := \mathcal{D}_{\text{enc}}(\mathbf{sk}_{\text{enc}}, \mathbf{C}_i)$. ii. If \mathcal{D}_{enc}'s output is not \perp, verify that: <ul style="list-style-type: none"> • $\mathbf{cm}_i^{\text{new}}$ equals $\text{COMM}_{s_i}^{\text{sn}}(v_i \parallel \text{COMM}_{r_i}^{\text{sn}}(a_{\text{pk}} \parallel \rho_i))$; • $\mathbf{sn}_i := \text{PRF}_{a_{\text{sk}}}^{\text{sn}}(\rho_i)$ does not appear on L. iii. If both checks succeed, output $\mathbf{c}_i := (\mathbf{addr}_{\text{pk}}, v_i, \rho_i, r_i, s_i, \mathbf{cm}_i^{\text{new}})$.
---	--

Figure 2: Construction of a DAP scheme using zk-SNARKs and other ingredients.

5.1 Instantiation of building blocks

We instantiate the DAP scheme construction from Section 4 (see Figure 2), aiming at a level of security of 128 bits. Doing so requires concrete choices, described next.

CRH, PRF, COMM from SHA256. Let \mathcal{H} be the SHA256 compression function, which maps a 512-bit input to a 256-bit output. We mostly rely on \mathcal{H} , rather than the “full” hash, since this suffices for our fixed-size single-block inputs, and it simplifies the construction of C_{POUR} (see Section 5.2). We instantiate CRH, PRF, COMM via \mathcal{H} (under suitable assumptions on \mathcal{H}).

First, we instantiate the collision-resistant function CRH as $\mathcal{H}(z)$ for $z \in \{0, 1\}^{512}$; this function compresses “two-to-one”, so it can be used to construct binary Merkle trees.¹⁴

Next, we instantiate the pseudorandom function $\text{PRF}_x(z)$ as $\mathcal{H}(x \| z)$, with $x \in \{0, 1\}^{256}$ as the seed, and $z \in \{0, 1\}^{256}$ as the input.¹⁵ Thus, the derived functions are:

$$\text{PRF}_x^{\text{addr}}(z) := \mathcal{H}(x \| 00 \| z), \quad \text{PRF}_x^{\text{sn}}(z) := \mathcal{H}(x \| 01 \| z), \quad \text{PRF}_x^{\text{pk}}(z) := \mathcal{H}(x \| 10 \| z) ,$$

with $x \in \{0, 1\}^{256}$ and $z \in \{0, 1\}^{254}$.

As for the commitment scheme COMM, we only use it in the following pattern:

$$\begin{aligned} k &:= \text{COMM}_r(a_{\text{pk}} \| \rho) , \\ \text{cm} &:= \text{COMM}_s(v \| k) . \end{aligned}$$

Due to our instantiation of PRF, a_{pk} is 256 bits. So we can set ρ also to 256 bits and r to $256 + 128 = 384$ bits; then we can compute

$$k := \text{COMM}_r(a_{\text{pk}} \| \rho) \text{ as } \mathcal{H}(r \| [\mathcal{H}(a_{\text{pk}} \| \rho)]_{128}) .$$

Above, $[\cdot]_{128}$ denotes that we are truncating the 256-bit string to 128 bits (say, by dropping least-significant bits, as in our implementation). Heuristically, for any string $z \in \{0, 1\}^{128}$, the distribution induced by $\mathcal{H}(r \| z)$ is 2^{-128} -close to uniform, and this forms the basis of the statistically-hiding property. For computing cm, we set coin values to be 64-bit integers (so that, in particular, $v_{\max} = 2^{64} - 1$ in our implementation), and then compute

$$\text{cm} := \text{COMM}_s(v \| k) \text{ as } \mathcal{H}(k \| 0^{192} \| v) .$$

Noticeably, above we are *ignoring* the commitment randomness s . The reason is that we already know that k , being the output of a statistically-hiding commitment, can serve as randomness for the next commitment scheme.

Instantiating the NP statement POUR. The above choices imply a concrete instantiation of the NP statement POUR (see Section 4.2). Specifically, in our implementation, POUR checks that the following holds, for each $i \in \{1, 2\}$:

- path_i is an authentication path for leaf cm_i^{old} with respect to root rt , in a CRH-based Merkle tree;
- $a_{\text{pk}, i}^{\text{old}} = \mathcal{H}(a_{\text{sk}, i}^{\text{old}} \| 0^{256})$;
- $\text{sn}_i^{\text{old}} = \mathcal{H}(a_{\text{sk}, i}^{\text{old}} \| 01 \| [\rho_i^{\text{old}}]_{254})$;
- $\text{cm}_i^{\text{old}} = \mathcal{H}(\mathcal{H}(r_i^{\text{old}} \| [\mathcal{H}(a_{\text{pk}, i}^{\text{old}} \| \rho_i^{\text{old}})]_{128}) \| 0^{192} \| v_i^{\text{old}})$;

¹⁴A single exception: we still compute h_{Sig} according to the full hash SHA256, rather than its compression function, because there is no need for this computation to be verified by C_{POUR} .

¹⁵This assumption is reminiscent of previous works analyzing the security of hash-based constructions (e.g., [Bel06]). However in this work we assume that a portion of the compression function is the *seed* for the pseudorandom function, rather than using the chaining variable as in [Bel06].

- $\text{cm}_i^{\text{new}} = \mathcal{H}(\mathcal{H}(r_i^{\text{new}} \parallel [\mathcal{H}(a_{\text{pk},i}^{\text{new}} \parallel \rho_i^{\text{new}})]_{128}) \parallel 0^{192} \parallel v_i^{\text{new}})$; and
- $h_i = \mathcal{H}(a_{\text{sk},i}^{\text{old}} \parallel 10 \parallel b_i \parallel [h_{\text{Sig}}]_{253})$ where $b_1 := 0$ and $b_2 := 1$.

Moreover, POUR checks that $v_1^{\text{new}} + v_2^{\text{new}} + v_{\text{pub}} = v_1^{\text{old}} + v_2^{\text{old}}$, with $v_1^{\text{old}}, v_2^{\text{old}} \geq 0$ and $v_1^{\text{old}} + v_2^{\text{old}} < 2^{64}$.

Finally, as mentioned, in order for C_{POUR} to be well-defined, we need to fix a Merkle-tree depth d_{tree} . In our implementation, we fix $d_{\text{tree}} = 64$, and thus support up to 2^{64} coins.

Instantiating Sig. For the signature scheme **Sig**, we use ECDSA to retain consistency and compatibility with the existing `bitcoind` source code. However, standard ECDSA is malleable: both (r, s) and $(r, -s)$ verify as valid signatures. We use a non-malleable variant, where s is restricted to the “lower half” of field elements. While we are not aware of a formal SUF-1CMA proof for this variant, its use is consistent with proposals to resolve Bitcoin transaction malleability [Wui14].¹⁶

Instantiating Enc. For the encryption scheme **Enc**, we use the key-private Elliptic-Curve Integrated Encryption Scheme (ECIES) [Cer00]; it is one of the few standardized key-private encryption schemes with available implementations.

5.2 Arithmetic circuit for pouring coins

Our DAP scheme construction from Section 4 (see Figure 2) also requires zk-SNARKs relative to the NP statement POUR. These are obtained by invoking a zk-SNARK for arithmetic circuit satisfiability (see Section 2.4) on an arithmetic circuit C_{POUR} , which verifies the NP statement POUR. In our instantiation, we rely on the implementation of [BCTV14] for the basic zk-SNARK (see Section 2.4), and apply it to the circuit C_{POUR} whose construction is described next.

5.2.1 An arithmetic circuit for verifying SHA256’s compression function

The vast majority of the “verification work” in POUR is verifying computations of \mathcal{H} , the compression function of SHA256 (see Section 5.1). Thus, we begin by discussing our construction of an arithmetic circuit $C_{\mathcal{H}}$ for verifying SHA256 computations. Later, in Section 5.2.2, we discuss the construction of C_{POUR} , given the circuit $C_{\mathcal{H}}$.

We wish to construct an arithmetic circuit $C_{\mathcal{H}}$ such that, for every 256-bit digest h and 512-bit input z , $(h, z) \in \mathcal{R}_{C_{\mathcal{H}}}$ if and only if $h = \mathcal{H}(z)$. Naturally, our goal is to minimize the size of $C_{\mathcal{H}}$. Our high-level strategy is to construct $C_{\mathcal{H}}$, piece by piece, by closely following the SHA256 official specification [Nat12]. For each subcomputation of SHA256, we use nondeterminism and field operations to verify the subcomputation using as few gates as possible.

Overview of SHA256’s compression function. The primitive unit in SHA256 is a 32-bit *word*. All subcomputations are simple word operations: three bitwise operations (`and`, `or`, `xor`), shift-right, rotate-right, and addition modulo 2^{32} . The compression function internally has a *state* of 8 words, initialized to a fixed value, and then transformed in 64 successive rounds by following the 64-word *message schedule* (deduced from the input z). The 256-bit output is the concatenation of the 8 words of the final state.

Representing a state. We find that, for each word operation (except for addition modulo 2^{32}), it is more efficient to verify the operation when its inputs are represented as separate wires, each carrying a bit. Thus, $C_{\mathcal{H}}$ maintains the 8-word state as 256 individual wires, and the 64-word message schedule as $64 \cdot 32$ wires.

Addition modulo 32. To verify addition modulo 2^{32} we use techniques employed in previous work [PGHR13, BCG⁺13, BCTV14]. Given two words A and B , we compute $\alpha := \sum_{i=0}^{31} 2^i (A_i + B_i)$.

¹⁶In practice, one might replace this ECDSA variant with an EC-Schnorr signature satisfying SUF-1CMA security with proper encoding of EC group elements; the performance would be similar.

Because \mathbb{F} has characteristic larger than 2^{33} , there is no wrap around; thus, field addition coincides with integer addition. We then make a non-deterministic guess for the 33 bits α_i of α (including carry), and enforce consistency by requiring that $\alpha = \sum_{i=0}^{32} 2^i \alpha_i$. To ensure that each $\alpha_i \in \{0, 1\}$, we use a 33-gate subcircuit computing $\alpha_i(\alpha_i - 1)$, all of which must be 0 for the subcircuit to be satisfiable. Overall, verifying addition modulo 2^{32} only requires 34 gates. This approach extends in a straightforward way to summation of more than two terms.

Verifying the SHA256 message schedule. The first 16 words W_i of the message schedule are the 16 words of the 512-bit input z . The remaining 48 words are computed as $W_t := \sigma_1(W_{t-2}) + W_{t-7} + \sigma_0(W_{t-15}) + W_{t-16}$, where $\sigma_0(W) := \text{rotr}_7(W) \oplus \text{rotr}_{18}(W) \oplus \text{shr}_3(W)$ and σ_1 has the same structure but different rotation and shift constants.

The rotation and shift amounts are constants, so rotates and shifts can be achieved by suitable wiring to previously computed bits (or the constant 0 for high-order bits in `shr`). Thus, since the XOR of 3 bits can be computed using 2 gates, both σ_0 and σ_1 can be computed in 64 gates. We then compute (or more precisely, guess and verify) the addition modulo 2^{32} of the four terms.

Verifying the SHA256 round function. The round function modifies the 8-word state by changing two of its words and then permuting the 8-word result.

Each of the two modified words is a sum modulo 2^{32} of (i) round-specific constant words K_t ; (ii) message schedule words W_t ; and (iii) words obtained by applying simple functions to state words. Two of those functions are bitwise *majority* ($\text{Maj}(A, B, C)_i = 0$ if $A_i + B_i + C_i \leq 1$ else 1) and bitwise *choice* ($\text{Ch}(A, B, C)_i = B_i$ if $A_i = 1$, else C_i). We verify correct computation of `Maj` using 2 gates per output bit, and `Ch` with 1.

Then, instead of copying 6 unchanged state words to obtain the permuted result, we make the permutation implicit in the circuit’s wiring, by using output wires of previous sub-computations (sometimes reaching 4 round functions back) as input wires to the current sub-computation.

Performance. Overall, we obtain an arithmetic circuit $C_{\mathcal{H}}$ for verifying SHA256’s compression function with less than 30 000 arithmetic gates. See Figure 3 for a breakdown of gate counts.

Gate count for $C_{\mathcal{H}}$	
Message schedule	8032
All rounds	19 584
1 round (of 64)	306
Finalize	288
Total	27 904

Figure 3: Size of circuit $C_{\mathcal{H}}$ for SHA256’s compression function.

Comparison with generic approaches. We constructed the circuit $C_{\mathcal{H}}$ from scratch. We could have instead opted for more generic approaches: implement SHA256’s compression function in a higher-level language, and use a circuit generator to obtain a corresponding circuit. However, generic approaches are significantly more expensive for our application, as we now explain.

Starting from the SHA256 implementation in PolarSSL (a popular cryptographic library) [Pol13], it is fairly straightforward to write a C program for computing \mathcal{H} . We wrote such a program, and gave it as input to the circuit generator of [PGHR13]. The output circuit had 58160 gates, more than twice larger than our hand-optimized circuit.

Alternatively, we also compiled the same C program to TinyRAM, which is the architecture supported in [BCG⁺13]; we obtained a 5371-instruction assembly code that takes 5704 cycles to execute on TinyRAM. We could then invoke the circuit generator in [BCG⁺13] when given this TinyRAM program and time bound. However, each TinyRAM cycle costs ≈ 1000 gates, so the resulting circuit would have at least $5.7 \cdot 10^6$ gates, i.e., over 190 times larger than our circuit. A

similar computation holds for the circuit generator in [BCTV14], which supports an even more flexible architecture.

Thus, overall, we are indeed much better off constructing $C_{\mathcal{H}}$ from scratch. Of course, this is not surprising, because a SHA256 computation is almost a “circuit computation”: it does not make use of complex program flow, accesses to memory, and so on. Thus, relying on machinery developed to support much richer classes of programs does not pay off.

5.2.2 Arithmetic circuit for POUR

The NP statement POUR requires verifying membership in a Merkle tree based on \mathcal{H} , a few additional invocations of \mathcal{H} , and integer addition and comparison. We construct the circuit C_{POUR} for POUR by combining various subcircuits verifying each of these. There remains to discuss the subcircuits for verifying membership in a Merkle tree (using the aforementioned subcircuit $C_{\mathcal{H}}$ for verifying invocations of \mathcal{H}), and integer addition and comparison.

Merkle tree membership. We need to construct an arithmetic circuit that, given a root rt , authentication path path , and coin commitment cm , is satisfied if and only if path is a valid authentication path for the leaf cm with respect to the root rt . The authentication path path includes, for each layer i , an auxiliary hash value h_i and a bit r_i specifying whether h_i was the left ($r_i = 0$) or the right ($r_i = 1$) child of the parent node. We then check membership in the Merkle tree by verifying invocations of \mathcal{H} , bottom-up. Namely, for $d = 64$, we set $k_{d-1} = \text{cm}$; then, for each $i = d-1, \dots, 1$, we set $B_i = h_i \| k_i$ if $r_i = 0$ else $k_i \| h_i$, and compute $k_{i-1} = \mathcal{H}(B_i)$. Finally we check that the root k_0 matches the given root rt .

Integer addition. We need to construct an arithmetic circuit that, given 64-bit integers A, B, C (presented as binary strings), is satisfied if and only if $C = A + B$ over the integers. Again relying on the fact that \mathbb{F} ’s characteristic is sufficiently large, we do so by checking that $\sum_{i=0}^{63} 2^i c_i = \sum_{i=0}^{63} 2^i (b_i + a_i)$ over \mathbb{F} ; this is enough, because there is no wrap around.

Integer comparison. We need to construct an arithmetic circuit that, given two 64-bit integers A, B (represented in binary), is satisfied if and only if $A + B$ fits in 64 bits (i.e. $A + B < 2^{64}$). We do so by checking that $\sum_{i=0}^{63} 2^i (b_i + a_i) = \sum_{i=0}^{63} c_i$ for some $c_i \in \{0, 1\}$. Indeed, if $A + B < 2^{64}$ then it suffices to take c_i as the binary representation of $A + B$. However, if $A + B \geq 2^{64}$ then no choice of c_i can satisfy the constraint as $\sum_{i=0}^{63} c_i \leq 2^{64} - 1$. Overall, this requires 65 gates (1 gate for the equality check, and 64 gates for ensuring that c_0, \dots, c_{63} are boolean).

Overall circuit sizes. See Figure 4 for the size of C_{POUR} . More than 99% of the gates are devoted to verifying invocations of \mathcal{H} .

Gate count for C_{POUR}	
Ensure cm_1^{old} is in Merkle tree (1 layer out of 64)	1 802 304 (28 161)
Ensure cm_2^{old} is in Merkle tree (1 layer out of 64)	1 802 304 (28 161)
Check computation of $\text{sn}_1^{\text{old}}, \text{sn}_2^{\text{old}}$	2×27904
Check computation of $a_{\text{pk},1}^{\text{old}}, a_{\text{pk},2}^{\text{old}}$	2×27904
Check computation of $\text{cm}_1^{\text{old}}, \text{cm}_2^{\text{old}}, \text{cm}_1^{\text{new}}, \text{cm}_2^{\text{new}}$	4×83712
Check computation of h_1, h_2	2×27904
Ensure that $v_1^{\text{new}} + v_2^{\text{new}} + v_{\text{pub}} = v_1^{\text{old}} + v_2^{\text{old}}$	1
Ensure that $v_1^{\text{old}} + v_2^{\text{old}} < 2^{64}$	65
Miscellaneous	2384
Total	4 109 330

Figure 4: Size of the circuit C_{POUR} , which verifies the statement POUR.

6 Integration with existing ledger-based currencies

Zerocash can be deployed atop any ledger (even one maintained by a central bank). Here, we briefly detail integration with the Bitcoin protocol. Unless explicitly stated otherwise, in the following section when referring to *Bitcoin*, and its unit of account *bitcoin* (plural bitcoins), we mean the underlying protocol and software, not the currency system. (The discussion holds, with little or no modification, for many forks of Bitcoin, also known as “altcoins”, such as Litecoin.)

By introducing new transaction types and payment semantics, Zerocash breaks compatibility with the Bitcoin network. While Zerocash could be integrated into Bitcoin (the actual currency and its supporting software) via a “flag day” where a super-majority of Bitcoin miners simultaneously adopt the new software, we neither expect nor advise such integration in the near future and suggest using Zerocash in a separate altcoin.

Integrating Zerocash into Bitcoin consists of adding a new transaction type, Zerocash transactions, and modifying the protocol and software to invoke Zerocash’s DAP interface to create and verify these transactions. There are at least two possible approaches to this integration. The first approach replaces all bitcoins with zerocoins, making all transactions anonymous at the cost of losing any additional Bitcoin functionality provided by, e.g., the Bitcoin scripting language (see Section 6.1). The second approach maintains this functionality, adding a *parallel* Zerocash currency, *zerocoins*, which can be converted to and from bitcoin at a one-to-one rate (see Section 6.2). Options for protocol-level modifications for the later approach are discussed in Section 6.3; the former can be readily inferred. In Section 6.4 we discuss anonymizing the network layer of Bitcoin and anonymity safeguards.

6.1 Integration by replacing the base currency

One approach is to alter the underlying system so that all monetary transactions are done using Zerocash, i.e., by invoking the DAP interface and writing/reading the associated transactions in the distributed ledger.

As seen in Section 3, this suffices to offer the core functionality of payments, minting, merging, splitting, etc., while assuring users that all transactions using this currency are anonymous. However, this has several drawbacks: (1) All pour transactions incur the cost of generating a zk-SNARK proof. (2) If Bitcoin supports additional features, such as a scripting language for specifying conditions for claiming bitcoins (as in Bitcoin), then these features are lost.¹⁷ (3) Bitcoin allows the flexibility of spending unconfirmed transactions; instead, with a Zerocash-only Bitcoin, this flexibility is lost: transactions must be confirmed before they can be spent. (And this imposes a minimal delay between receiving funds and spending them.)

6.2 Integration by hybrid currency

A different approach is to extend Bitcoin with a parallel, anonymized currency of “zerocoins”, existing alongside bitcoins, using the same ledger, and with the ability to convert freely between the two. The behavior and functionality of regular bitcoins is unaltered; in particular, they may support functionality such as scripting.

In this approach, the Bitcoin ledger consists of Bitcoin-style transactions, containing inputs and outputs [Nak09]. Each input is either a pointer to an output of a previous transaction (as in plain Bitcoin), or a Zerocash pour transaction (which contributes its *public* value, v_{pub} , of bitcoins to this transaction). Outputs are either an amount and destination public address/script (as in plain

¹⁷However, in principle POUR could be extended to include a scripting language interpreter.

Bitcoin), or a Zerocash mint transaction (which consumes the input bitcoins to produce zerocoins). The usual invariant over bitcoins is maintained and checked in plain view: the sum of bitcoin inputs (including pours' v_{pub}) must be at least the sum of bitcoin outputs (including mints' v), and any difference is offered as a transaction fee. However, the accounting for zerocoins consumed and produced is done separately and implicitly by the DAP scheme.

The life cycle of a zerocoins is as follows.

Creating new zerocoins. A mint transaction consumes v worth of bitcoins as inputs, and outputs coin commitment worth v zerocoins. The v bitcoins are effectively destroyed, in exchange for the newly-minted zerocoins.

Spending zerocoins. Zerocoins can then be transferred, split, and merged into other zerocoins arbitrarily, via pour transactions which, instead of explicit inputs, include zero-knowledge proofs that such inputs exist. Pour transactions may optionally reveal a non-zero public output v_{pub} . This is either left unclaimed as a transaction fee,¹⁸ placed into a standard Bitcoin transaction output (e.g., one paying to a public key) or consumed by a mint transaction. Thus, v_{pub} bitcoins are created ex nihilo (similarly to how coinbase transactions produce bitcoin outputs as mining reward), in exchange for destroying that amount of zerocoins. The Bitcoin outputs must be included in the transaction string `info`, which is included as part of a pour transaction; transaction non-malleability ensures that all this information is bound together.

Spending multiple zerocoins. To allow for pours to span more than two input and output coins, `txPour` structures may be chained together within one transaction by marking some output coin commitments as intermediates and having subsequent pours in the same transaction constructed relative to an ephemeral Merkle tree consisting of only the intermediates commitments. For example, a transaction might accept four input coins, with the first two `Pour` operations combining two of the inputs to produce an intermediate commitment each and a final `Pour` combining the two intermediate commitments into a final output new coin. Since the intermediate results are consumed instantly within the transaction, they need not be recorded in the global Merkle tree or have their serial numbers marked as spent.

Transaction fees. Collecting transaction fees is done as usual, via a coinbase transaction added to each block, which pays as mining reward the difference between the total inputs (bitcoin and pours' v_{pub}) and total outputs (bitcoin and mints' v) in this block. Payment is either in bitcoins or in newly-minted zerocoins (via a `Mint`).

Validation and block generation. All transactions are verified via `VerifyTransaction` when they are received by a node. Any plain Bitcoin inputs and outputs are processed as usual, and any Zerocash inputs and outputs are checked using `VerifyTransaction` with the entire Bitcoin transaction fed in as `info` for authentication. Once these transactions are assembled into a candidate block, each transaction needs to be verified again to ensure its serial number has not become spent or its Merkle root invalid. If these checks pass, the set of new coin commitments and spent serial numbers output by the included transactions are added to the global sets, and the new Merkle root and a digest of the serial number list is stored in the new block.¹⁹ Embedding this data simplifies statekeeping and allows nodes to readily verify they have the correct coin list and serial number list. Upon receiving a candidate block, nodes validate the block is formed correctly with respect to the above procedure.

Receiving payments. In order to receive payments to an address, users may scan the block chain by running the `Receive` on every pour transaction. Alternatively they may receive coin information

¹⁸Since transaction fees may potentially be claimed by any node in the network, they represent the sole zerocoins output that cannot be hidden from public view even in a Zerocash-only system.

¹⁹This can be stored in the coinbase transaction, as certain other data currently is, or in a new field in the block header.

via some out-of-band mechanism (e.g., via encrypted email). The former process is nearly identical to the one proposed for “stealth addresses” for Bitcoin. In the worst case, scanning the block chain requires a trial decryption of every ciphertext C . We expect many scenarios to provide explicit notification, e.g., in interactive purchases where a communication channel already exists from the payer to the payee. (Implementations may opt to drop the receive mechanism entirely, and require out-of-band notification, in order to avoid storing the ciphertexts in the block chain.)

6.3 Extending the Bitcoin protocol to support the combined semantics

While the section above describes the life-cycle of a zerocoins and semantics of the system, there remains the question of how transactions acquire the above necessary semantics. Two implementation approaches are possible, with different engineering tradeoffs.

The first approach is to extend the protocol and its implementation with hard-coded validation of Zerocash transactions, reading them from new, designated fields in transactions and running `VerifyTransaction`. In this case the zk-SNARK itself effectively replaces the scripting language for Zerocash transactions.

The second approach is to extend Bitcoin’s scripting language by adding an opcode that invokes `VerifyTransaction`, with the requisite arguments embeded alongside the opcode script. Such transactions must be exempt from the requirement they reference an input (as they are Zerocash transactions are self-contained), and, like coinbase transactions, be able to create bitcoins ex nihilo (to account for v_{pub}). Moreover, while `VerifyTransaction` is run at the standard point in the Bitcoin transaction processing flow for evaluating scripts, the coin commitments and spent serial numbers are not actually added to `CMList` (resp., `SNList`) until their containing block is accepted (i.e., merely verifying a transaction does not have side effects).

6.4 Additional anonymity considerations

Zerocash only anonymizes the transaction ledger. Network traffic used to announce transactions, retrieve blocks, and contact merchants still leaks identifying information (e.g., IP addresses). Thus users need some anonymity network to safely use Zerocash. The most obvious way to do this is via Tor [DMS04]. Given that Zerocash transactions are not low latency themselves, Mixnets (e.g., Mixminion [DDM03]) are also a viable way to add anonymity (and one that, unlike Tor, is not as vulnerable to traffic analysis). Using mixnets that provide email-like functionality has the added benefit of providing an out-of-band notification mechanism that can replace `Receive`.

Additionally, although in theory all users have a single view of the block chain, a powerful attacker could potentially fabricate an additional block *solely* for a targeted user. Spending any coins with respect to the updated Merkle tree in this “poison-pill” block will uniquely identify the targeted user. To mitigate such attacks, users should check with trusted peers their view of the block chain and, for sensitive transactions, only spend coins relative to blocks further back in the ledger (since creating the illusion for multiple blocks is far harder).

7 Experiments

To measure the performance of Zerocash, we ran several experiments. First, we benchmarked the performance of the zk-SNARK for the NP statement `POUR` (Section 7.1) and of the six DAP scheme algorithms (Section 7.2). Second, we studied the impact of a higher block verification time via a simulation of a Bitcoin network (Section 7.3).

7.1 Performance of zk-SNARKs for pouring coins

Our zk-SNARK for the NP statement POUR is obtained by constructing an arithmetic circuit C_{POUR} for verifying POUR , and then invoking the generic implementation of zk-SNARK for arithmetic circuit satisfiability of [BCTV14] (see Section 2.4). The arithmetic circuit C_{POUR} is built from scratch and hand-optimized to exploit nondeterministic verification and the large field characteristic (see Section 5.2).

Figure 5 reports performance characteristics of the resulting zk-SNARK for POUR . This includes three settings: single-thread performance on a laptop machine; and single-thread and multi-thread performance on a desktop machine. (The time measurements are the average of 10 runs, with standard deviation under 2.5%.) For instance, with single-thread code on the laptop machine, we obtain that:

- Key generation takes 7 min 48 s, and results in a proving key pk_{POUR} of 896 MiB and a verification key vk_{POUR} of 749 B. This is performed only once, as part of the `Setup` algorithm.
- Producing a proof π_{POUR} requires about 3 minutes; proofs have a constant size of 288 B. Proof generation is a subroutine of the `Pour` algorithm, and the resulting proof is included in the corresponding pour transaction.
- A proof π_{POUR} can be verified in only 8.5 ms. Proof verification is a subroutine of the `VerifyTransaction` algorithm, when it is given as input a pour transaction to verify.

		Intel Core i7-2620M @ 2.70GHz 12GB of RAM	Intel Core i7-4770 @ 3.40GHz 16GB of RAM	
		1 thread	1 thread	4 threads
KeyGen	Time	7 min 48 s	5 min 11 s	1 min 47 s
	Proving key	896 MiB		
	Verification key	749 B		
Prove	Time	2 min 55 s	1 min 59 s	46 s
	Proof	288 B		
Verify	Time	8.5 ms	5.4 ms	

Figure 5: Performance of our zk-SNARK for the NP statement POUR . ($N = 10$, $\sigma \leq 2.5\%$)

7.2 Performance of Zerocash algorithms

In Figure 6 we report performance characteristics for each of the six DAP scheme algorithms in our implementation (single-thread on our desktop machine). For `VerifyTransaction`, we separately report the cost of verifying mint and pour transactions and, in the latter case, we exclude the cost of scanning L (e.g., to check if a serial number is duplicate);²⁰ for the case of `Receive`, we report the cost to process a given pour transaction in L .

We obtain that:

- `Setup` takes about 5 minutes to run; its running time is dominated by the running time of `KeyGen` on C_{POUR} . (Either way, `Setup` is run only once.) The size of the resulting public parameters pp is dominated by the size of pk_{POUR} .
- `CreateAddress` takes 326.0 ms to run. The size of the resulting address key pair is just a few hundred bytes.

²⁰Naturally, if SNList has 2^{64} serial numbers (the maximum possible in our implementation), then scanning is very expensive! However, we do not expect that a system like Zerocash will grow to 2^{64} transactions. Still, such a system may grow to the point that scanning SNList is too expensive. We detail possible mitigations to this in Section 8.3.2.

- Mint takes $23\ \mu\text{s}$ to run. It results in a coin of size $463\ \text{B}$ and mint transaction of size $72\ \text{B}$.
- Pour takes about 2 minutes to run. Besides Setup, it is the only “expensive” algorithm to run; as expected, its running time is dominated by the running time of Prove. For a transaction string info , it results in (two new coins and) a pour transaction of size $996\ \text{B} + |\text{info}|$.
- VerifyTransaction takes $8.3\ \mu\text{s}$ to verify a mint transaction and $5.7\ \text{ms}$ to verify a pour transaction; the latter’s time is dominated by that of Verify, which checks the zk-SNARK proof π_{POUR} .
- Receive takes $1.6\ \text{ms}$ per pour transaction.

Note that the above numbers do not include the costs of maintaining the Merkle tree because doing so is not the responsibility of the DAP scheme algorithms. Nevertheless, these additional costs are not large: (i) each update of the root of the CRH-based Merkle tree only requires d_{tree} invocations of CRH, and (ii) an authentication path consists of only d_{tree} digests of CRH. In our implementation, where $\text{CRH} = \mathcal{H}$ (the SHA256 compression function) and $d_{\text{tree}} = 64$, each update requires 64 invocations of \mathcal{H} and an authentication path requires $64 \cdot 32\ \text{B} = 2\ \text{KiB}$ of storage.

Remark. If one does not want to rely on the ledger to communicate coins, via the ciphertexts $\mathbf{C}_1, \mathbf{C}_2$, and instead rely instead on some out-of-band mechanism (e.g., encrypted email), then the Receive algorithm is not needed, and moreover, many of the aforementioned sizes decrease because some pieces of data are not needed anymore; we denoted these pieces of data with “ \star ” in Figure 6. (E.g., the size of an address key pair is reduced to only $64\ \text{B}$, and the size of a coin to only $120\ \text{B}$.)

7.3 Large-scale network simulation

Because Bitcoin mining typically takes place on dedicated GPUs or ASICs, the CPU resources to execute the DAP scheme algorithms are often of minimal consequence to network performance. There is one potential exception to this rule: the VerifyTransaction algorithm must be run by all of the network nodes in the course of routine transaction validation. The time it takes to perform this verification may have significant impact on network performance.

In the Zerocash implementation (as in Bitcoin), every Zerocash transaction is verified at each hop as it is forwarded through the network and, potentially, again when blocks containing the transaction are verified. Verifying a block consists of checking the proof of work and validating the contained transactions. Thus Zerocash transactions may take longer to spread through the network and blocks containing Zerocash transactions may take longer to verify. While we are concerned with the first issue, the potential impact of the second issue is cause for greater concern. This is because Zerocash transactions cannot be spent until they make it onto the ledger.

Because blocks are also verified at each hop before they are forwarded through the network, delays in block verification slow down the propagation of new blocks through the network. This causes nodes to waste CPU-cycles mining on out-of-date blocks, reducing the computational power of the network and making it easier to mount a “51% attack” (dishonest majority of miners) on the distributed ledger.

It is a priori unclear whether this potential issue is a real concern. Bitcoin caches transaction verifications, so a transaction that was already verified when it propagated through the network need not be verified again when it is seen in a block. The unknown is what percentage of transactions in a block are actually in any given node’s cache. We thus conduct a simulation of the Bitcoin network to investigate both the time it takes Zerocash transactions to make it onto the ledger and establish the effects of Zerocash transactions on block verification and propagation. We find that Zerocash transactions can be spent reasonably quickly and that the effects of increased block validation time are minimal.

Intel Core i7-4770 @ 3.40GHz 16GB of RAM		
1 thread		
Setup	Time	5 min 17 s
	Size of pp	896 MiB
CreateAddress	size of pk_{POUR}	896 MiB
	size of vk_{POUR}	749 B
Mint	* size of pp_{enc}	0 B
	size of pp_{sig}	0 B
Pour	Time	326.0 ms
	Size of addr_{pk}	343 B
VerifyTransaction	size of a_{pk}	32 B
	* size of pk_{enc}	311 B
Receive	Size of addr_{sk}	319 B
	size of a_{sk}	32 B
Receive	* size of sk_{enc}	287 B
	Time	23 μ s
Mint	Size of coin \mathbf{c}	463 B
	size of addr_{pk}	343 B
Mint	size of v	8 B
	size of ρ	32 B
Mint	size of r	48 B
	size of s	0 B
Mint	size of cm	32 B
	Size of tx_{Mint}	72 B
Mint	size of cm	32 B
	size of v	8 B
Mint	size of k	32 B
	size of s	0 B
Pour	Time	2 min 2.01 s
	Size of tx_{Pour}	996 B + $ \text{info} $
Pour	size of rt	32 B
	size of $\text{sn}_1^{\text{old}}, \text{sn}_2^{\text{old}}$	2×32 B
Pour	size of $\text{cm}_1^{\text{new}}, \text{cm}_2^{\text{new}}$	2×32 B
	size of v_{pub}	8 B
Pour	size of info	$ \text{info} $
	size of pk_{sig}	66 B
Pour	size of h_1, h_2	2×32 B
	size of π_{POUR}	288 B
Pour	* size of $\mathbf{C}_1, \mathbf{C}_2$	2×173 B
	size of σ	64 B
VerifyTransaction	Time for mint tx	8.3 μ s
	Time for pour tx (excludes L scan)	5.7 ms
Receive	Time (per pour tx)	1.6 ms

Figure 6: Performance of Zerocash algorithms. Above, we report the sizes of pp_{enc} and pp_{sig} as 0 B, because these parameters are “hardcoded” in the libraries we rely on for Enc and Sig. ($N = 10$ with $\sigma \leq 2.5\%$ for all except that, due to variability at short timescales, $\sigma(\text{Mint}) \leq 3.3 \mu\text{s}$ and $\sigma(\text{VerifyTransaction}) \leq 1.9 \mu\text{s}$)

Simulation design. Because Zerocash requires breaking changes to the Bitcoin protocol, we cannot test our protocol in the live Bitcoin network or even in the dedicated testnet. We must run our own private testnet. For efficiency and cost reasons, we would like to run as many Bitcoin nodes as possible on the least amount of hardware. This raises two issues. First, reducing the proof of work to practical levels while still preserving a realistic rate of new blocks is difficult (especially on

virtualized hardware with variable performance). Second, the overhead of zk-SNARK verification prevents us from running many Bitcoin nodes on one virtualized server.

The frequency of new blocks can be modeled as a Poisson process with a mean of Λ_{block} seconds.²¹ To generate blocks stochastically, we modify `bitcoind` to fix its block difficulty at a trivial level²² and run a Poisson process, on the simulation control server, which trivially mines a block on a randomly selected node. This preserves the distribution of blocks, without the computational overhead of a real proof of work. Another Poisson process triggering mechanism, with a different mean Λ_{tx} , introduces new transactions at random network nodes.

To differentiate which transactions represent normal Bitcoin expenditures versus which contain Zerocash pour transactions, simulated Zerocash transactions pay a unique amount of bitcoins (we set this value arbitrarily at 7 BTC). If a transaction’s output matches this preset value, and it is not in verification cache, then our modified Bitcoin client inserts a 10 ms delay simulating the runtime of `VerifyTransaction`.²³ Otherwise transactions are processed as specified by the Bitcoin protocol. We vary the amount of simulated Zerocash traffic by varying the number of transactions with this particular output amount. This minimizes code changes and estimates only the generic impact of verification delays and not of any specific implementation choice.

Methodology. Recent research [DW13] suggests that the Bitcoin network contains 16,000 distinct nodes though most are likely no longer participating: approximately 3,500 are reachable at any given time. Each node has an average of 32 open connections to randomly selected peers. As of November 2013, the peak observed transaction rate for Bitcoin is slightly under one transaction per second [Lee13].

In our simulation, we use a 1000-node network in which each node has an average of 32 peers, transactions are generated with a mean of $\Lambda_{\text{tx}} = 1$ s, a duration of 1 hour, and a variable percentage ϵ of Zerocash traffic. To allow for faster experiments, instead of generating a block every 10 minutes as in Bitcoin, we create blocks at an average of every $\Lambda_{\text{block}} = 150$ s (as in Litecoin, a popular altcoin).

We run our simulation for different traffic mixes, where ϵ indicates the percentage of Zerocash transactions and $\epsilon \in \{0\%, 25\%, 50\%, 75\%, 100\%\}$. Each simulation is run on 200 Amazon EC2 general-purpose `m1.medium` instances, in one region on a `10.10./16` private network. On each instance, we deploy 5 instances of `bitcoind`.²⁴

Results. Transactions are triggered by a blocking function call on the simulation control node that must connect to a random node and wait for it to complete sending a transaction. Because the Poisson process modeling transactions generates delays between such calls and not between the exact points when the node actually sends the transactions, the actual transaction rate is skewed. In our experiments the real transaction rate shifts away from our target of one per second to an average of one every 1.4 seconds.

In Figure 7 we plot three metrics for $\epsilon \in \{0\%, 25\%, 50\%, 75\%, 100\%\}$. Each is the average defined over the data from the entire run of the simulation for a given ϵ (i.e., they include multiple transactions and blocks).²⁵ *Transaction latency* is the interval between a transaction’s creation and

²¹Since computational power is added to the Bitcoin network faster than the 2-week difficulty adjustment period, the frequency of block generation is actually skewed. As our experiments run for at most an hour, we ignore this.

²²These code modifications have been rendered moot by the subsequent inclusion of a “regtest” mode in Bitcoin 0.9 that allows for precisely this type of behavior and block generation on command. At the time of our experiments, this feature was not available in a stable release. Future work should use this feature.

²³We used a generous delay of 10 ms (higher than the time reported in Figure 6) to leave room for machines slower than our desktop machine.

²⁴Higher densities of nodes per VM resulted in issues initializing all of the `bitcoind` instances on boot.

²⁵Because our simulated Bitcoin nodes ran on shared EC2 instances, they were subject to variable external load,

its inclusion in a block. *Block propagation time* comes in two flavors: (1) the average time for a new block to reach a node computed over the times for all nodes, and (2) the same average computed over only the last node to see the block.

Block verification time is the average time, over all nodes, required to verify a block. If verification caching was not effective, we would expect to see a marked increase in both block verification time and propagation time. Since blocks occur on average every 150 s, and we expect approximately one transaction each second, we should see $150 \times 10 \text{ ms} = 1500 \text{ ms}$ of delay if all transactions were non-cached Zerocash transactions. Instead, we see worst case 80 ms and conclude caching is effective. This results in a negligible effect on block propagation (likely because network operations dominate).

The time needed for a transaction to be confirmed, and hence spendable, is roughly 190 s. For slower block generation rates (e.g., Bitcoin’s block every 10 minutes) this should mean users must wait only one block before spending received transactions.

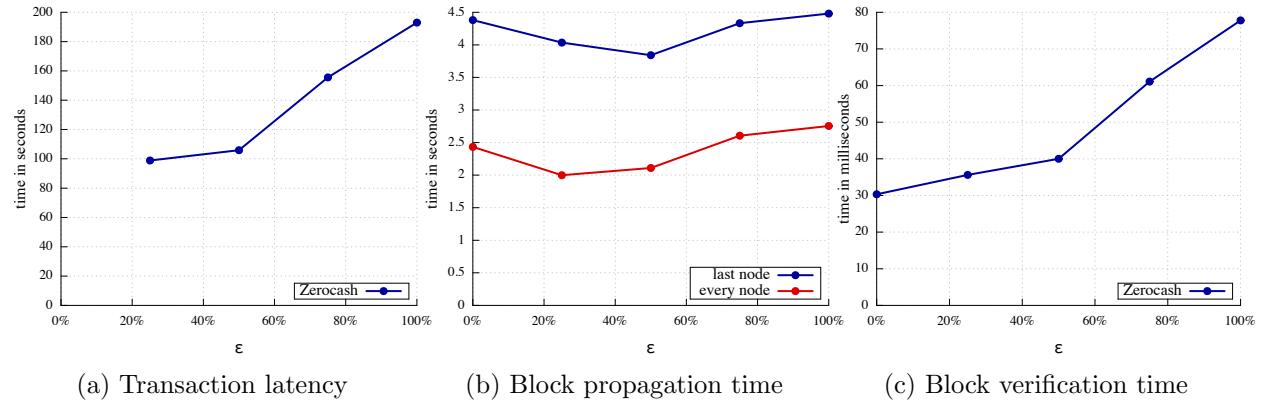


Figure 7: The average values of the three metrics we study, as a function of ϵ , the percentage of transactions that are Zerocash transactions. Note that, in (a), latency is undefined when $\epsilon = 0$ and hence omitted.

8 Optimizations and extensions

We outline several optimizations and extensions to Zerocash: everlasting anonymity (Section 8.1), faster block propagation (Section 8.2), and improved storage requirements (Section 8.3).

8.1 Everlasting anonymity

Since transactions may persist virtually forever on the ledger, users may wish to ensure the anonymity of their transactions also lasts forever, even if particular primitives are eventually broken (by cryptanalytic breakthrough, engineering progress, or quantum computers). As we now explain, the DAP scheme construction described in Section 4 is only computationally private, but can be modified to achieve *everlasting anonymity*.

Recall that every Pour operation publishes a pour transaction $\text{tx}_{\text{Pour}} = (\text{rt}, \text{sn}_1^{\text{old}}, \text{sn}_2^{\text{old}}, \text{cm}_1^{\text{new}}, \text{cm}_2^{\text{new}}, v_{\text{pub}}, \text{info}, *)$, where $* = (\text{pk}_{\text{sig}}, h_1, h_2, \pi_{\text{POUR}}, \mathbf{C}_1, \mathbf{C}_2, \sigma)$ and $\mathbf{C}_i = \mathcal{E}_{\text{enc}}(\text{pk}_{\text{enc},i}^{\text{new}}, (v_i^{\text{new}}, \rho_i^{\text{new}}, r_i^{\text{new}}, s_i^{\text{new}}))$. Observe that:

limiting the benchmark precision. Still, it clearly demonstrates that the mild additional delay does not cause catastrophic network effects.

- Since $h_{\text{Sig}} = \text{CRH}(\text{pk}_{\text{sig}})$ and $h_i = \text{PRF}_{a_{\text{sk},i}^{\text{old}}}^{\text{pk}}(h_{\text{Sig}})$, an unbounded adversary \mathcal{A} can iterate over all x until $\text{PRF}_x^{\text{pk}}(h_{\text{Sig}})$ equals h_i ; with overwhelming probability, there is only one such x , in which case it equals $a_{\text{sk},i}^{\text{old}}$. Thus, \mathcal{A} learns $a_{\text{sk},i}^{\text{old}}$, and hence $a_{\text{pk},i}^{\text{old}} := \text{PRF}_{a_{\text{sk},i}^{\text{old}}}^{\text{addr}}(0)$. This identifies the sender.
- An unbounded \mathcal{A} can also decrypt \mathbf{C}_i , so to learn $(v_i^{\text{new}}, \rho_i^{\text{new}}, r_i^{\text{new}}, s_i^{\text{new}})$; then, \mathcal{A} can try all possible x until $\text{COMM}_{s_i^{\text{new}}}(v_i^{\text{new}} \| \text{COMM}_{r_i^{\text{new}}}(\text{PRF}_x^{\text{addr}}(0) \| \rho_i^{\text{new}}))$ equals cm_i^{new} ; with overwhelming probability, there is only one such x , in which case it equals $a_{\text{sk},i}^{\text{new}}$. This identifies the recipient.

The above attacks can be prevented as follows. First, every sender must use any given address *only once* (for receiving or sending coins): after receiving a coin \mathbf{c} , a user u should immediately generate a new address and pour \mathbf{c} into a fresh one \mathbf{c}' relative to the new address; only afterwards can u spend the coin. Second, a user should not put any data in a ciphertext \mathbf{C}_i to communicate a coin’s information, but must instead use some (informationally-secure) out-of-band channel to do so. With these modifications (and recalling that COMM is statistically hiding and π_{POUR} is a perfect-zero-knowledge proof), one can verify that the pour transaction tx_{POUR} is statistically hiding, i.e., leaks no information even to unbounded adversaries.²⁶

8.2 Fast block propagation

As mentioned in Section 7.3, the higher block-verification time of Zerocash compared to, e.g., Bitcoin does not affect much block propagation. Even so, we note a simple modification that further mitigates concerns. Upon receiving a block, a node validates the proof of work and (optionally) transactions other than mint and pour, and then forward the block right away. Only afterwards, the node executes `VerifyTransaction` on any mint/pour transactions, before accepting it for use in transacting. Thus, blocks are still validated by every node (so the security properties are unhampered), and propagation delays in the broadcast of blocks are reduced.

In principle, this opens the possibility of a denial-of-service attack, in which the network is spammed with invalid blocks which pass the proof-of-work check but contain invalid mint or pour transactions. However, this attack appears unrealistic given the enormous (by design) cost of creating blocks passing the proof-of-work check.

8.3 Improved storage requirements

Beyond the ledger L , users need to maintain two lists: CMList , the list of all coin commitments, and SNList , the list of all serial numbers of spent coins (see Section 3.1). In our construction, CMList is required to deduce authentication paths to create new pour transactions (via `Pour`), while SNList is used to verify pour transactions (via `VerifyTransaction`). As the ledger grows, both CMList and SNList grow in size, and can eventually impose substantial storage requirements (though both are derived from, and smaller than, the block chain per se). We now explain how these storage requirements can be mitigated, by relying on smaller representations of CMList and SNList that suffice within our construction.

8.3.1 Supporting many coin commitments

To execute the `Pour` algorithm to spend a coin \mathbf{c} , a user u needs to provide an authentication path from \mathbf{c} ’s coin commitment to rt , the Merkle-tree root over CMList . If we make the following protocol modifications, u does not need all of CMList to compute this authentication path.

²⁶As for mint transactions, one can verify that they are already statistically hiding, without any modifications.

In each block B of transactions, we store the Merkle-tree path path_B from the first coin commitment in B to the root rt_B of the Merkle tree over CMList when the last block in the ledger is B . (In Zerocash, the additional per-block storage cost to store this information is only 2 KiB.)

Note that, given a block B and its successor block B' , the corresponding authentication paths path_B and $\text{path}_{B'}$ can be easily checked for consistency as follows. Let CMList_B and $\text{CMList}_{B'}$ be the two lists of coin commitments corresponding to the two ledgers ending in block B and B' respectively; since CMList_B (i.e., coin commitments to “to the left” of path_B) is a prefix of $\text{CMList}_{B'}$, $\text{path}_{B'}$ can be computed from path_B and B in time $O(|B|d_{\text{tree}})$, where d_{tree} is the tree depth.

When the user u first receives (or mints) the coin \mathbf{c} , and its coin commitment is included in a block B , u immediately computes path_B , by using the predecessor block and its authentication path. Afterwards, each time a new block is added to the ledger, u obtains a new path for \mathbf{c} by using the new block and the old path for \mathbf{c} . Thus, u only needs to act each time a new block is added, and each such update costs $O(d_{\text{tree}})$ per transaction in the block.

Overall, u incurs a storage requirement of only $O(d_{\text{tree}})$ for each coin he owns, and does not need to store CMList anymore.

8.3.2 Supporting many spent serial numbers

To execute the `VerifyTransaction` algorithm on a pour transaction tx_{Pour} , a user u needs access to SNList (in order to check for duplicate serial numbers). Note, in Bitcoin, nodes need to maintain only the list of unspent transaction outputs, which is pruned as outputs are spent. In a DAP scheme, in contrast, nodes have to maintain SNList , which is a list that *always grows*. We now explain how to mitigate this storage requirement, in three incremental steps.

Step 1. The first step is to build a Merkle tree over SNList so to allow easy-to-verify non-membership proofs for SNList ; this can be done by letting the leaves of the Merkle tree be the intervals of unspent serial numbers. Then, given the root rt of such tree, a serial number sn claimed to be unspent, and an authentication path path for an interval I , the user can check that path is valid for rt and that sn lies in I ; the root rt and path path would be part of the pour transaction tx_{Pour} to be verified. The problem with this approach, however, is that generating path (and also updating rt) requires knowledge of all of SNList .

Step 2. Next, instead of maintaining SNList in a single Merkle tree, we divide SNList , maintaining its chronological order, into sublists of serial numbers $\text{SNList}_0, \text{SNList}_1, \dots$ and build a Merkle tree over the intervals induced by each sublist (i.e., apply Step 1 to each sublist). This modification implies a corresponding modification for the auxiliary information stored in a pour transaction that allows `VerifyTransaction` to check it. Now, however, producing such auxiliary information is less expensive. Indeed, a user with a coin \mathbf{c} should maintain a list of authentication paths $\text{path}_{\mathbf{c},0}, \text{path}_{\mathbf{c},1}, \dots$ (one for each sublist). Only the last path, corresponding to the active sublist, needs to be updated when a serial number is added; the other sublists and authentication paths remain unchanged (and these old sublists can in fact be discarded). When the user spends the coin, he can simply include these paths in the pour transaction. While updating these paths is an efficient operation, computing the initial paths for \mathbf{c} is not, as it still requires the full set of sublists.

Step 3. To enable users to avoid the initial cost of computing paths for a new coin, we proceed as follows. First, a coin \mathbf{c} is extended to contain a time stamp $T_{\mathbf{c}}$ corresponding to when \mathbf{c} is created (minted or poured into); the coin’s commitment is modified to depend on the timestamp, and the timestamp is included in the clear within the transaction that creates the coin. Then, a user, upon spending \mathbf{c} , produces a zk-SNARK for the following NP statement: “for each Merkle-tree root created (or updated) after $T_{\mathbf{c}}$ there is an interval and an authentication path for that interval

such that the serial number of c is in that interval”. Depending on the number of Merkle trees in such an NP statement, such proofs may already be more efficient to produce, compared to the naive (Step 1) solution, using existing zk-SNARK implementations.

9 Concurrent work

Danezis et al. [DFKP13] suggest using zk-SNARKs to reduce proof size and verification time in Zerocoin. Our work differs from [DFKP13] in both supported functionality and scalability.

First, [DFKP13]’s protocol, like Zerocoin, only supports fixed-value coins, and is best viewed as a decentralized mix. Instead, we define, construct, and implement a full-fledged decentralized electronic currency, which provides anonymous payments of any amount.

Second, in [DFKP13], the complexity of the zk-SNARK generator, prover, and verifier all scale superlinearly in the number of coins, because their arithmetic circuit computes, *explicitly*, a product over all coins. In particular, the number of coins “mixed together” for anonymity cannot be large. Instead, in our construction, the respective complexities are polylogarithmic, polylogarithmic, and constant in the number of coins; our approach supports a practically-unbounded number of coins.

While we do not rely on Pedersen commitments, our approach also yields statistical (i.e., everlasting) anonymity; see the discussion in Section 8.1.

10 Conclusion

Decentralized currencies should ensure a user’s privacy from his peers when conducting legitimate financial transactions. Zerocash provides such privacy protection, by hiding user identities, transaction amounts, and account balances from public view. This, however, may be criticized for hampering accountability, regulation, and oversight. Yet Zerocash need not be limited to enforcing the basic monetary invariants of a currency system. The underlying zk-SNARK cryptographic proof machinery is flexible enough to support a wide range of policies. It can, for example, let a user prove that he paid his due taxes on all transactions *without* revealing those transactions, their amounts, or even the amount of taxes paid. As long as the policy can be specified by efficient nondeterministic computation using NP statements, it can (in principle) be enforced using zk-SNARKs, and added to Zerocash. This can enable automated, privacy-preserving verification and enforcement of a wide range of compliance and regulatory policies that would otherwise be invasive to check directly or might be bypassed by corrupt authorities. This raises research, policy, and engineering questions regarding which such policies are desirable and practically realizable.

Another research question is what new functionality can be realized by augmenting the capabilities already present in Bitcoin’s scripting language with zk-SNARKs that allow fast verification of expressive statements.

Acknowledgments

We thank Amazon for their assistance and kind donation of EC2 resources, and Gregory Maxwell for his advice regarding the Bitcoin codebase. We thank Iddo Ben-Tov and the SCIPR Lab members — Daniel Genkin, Lior Greenblatt, Shaul Kfir, Gil Timnat, and Michael Riabzev — for inspiring discussions. We thank Sharon Kessler for editorial advice.

This work was supported by: Amazon.com through an AWS in Education research grant; the Broadcom Foundation and Tel Aviv University Authentication Initiative; the Center for Science of Information (CSoI), an NSF Science and Technology Center, under grant agreement CCF-0939370; the Check Point Institute for Information Security; the U.S. Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL) under contract FA8750-11-2-0211; the European Community’s Seventh Framework Programme (FP7/2007-2013) under grant agreement number 240258; the Israeli Centers of Research Excellence I-CORE program (center 4/11); the Israeli Ministry of Science and Technology; the Office of Naval Research under contract N00014-11-1-0470; the Simons Foundation, with a Simons Award for Graduate Students in Theoretical Computer Science; and the Skolkovo Foundation with agreement dated 10/26/2011.

The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

A Overview of Bitcoin and Zerocoin

We provide an overview of the Bitcoin and Zerocoin protocols. For more details, we refer the reader to Nakamoto [Nak09] and Miers et al. [MGGR13] respectively.

A.1 Bitcoin

Bitcoin [Nak09] is a decentralized currency operated by a collection of mutually-distrusting peers. It consists of three basic components: (i) a peer-to-peer network for broadcasting new transactions; (ii) semantics for identifying and validating new transactions; and (iii) a protocol for maintaining a decentralized ledger, known as the *block chain*, that stores the history of all valid transactions so far.

Identities in Bitcoin are represented via ECDSA public keys. Each user u generates an ECDSA key pair $(\text{vk}_u, \text{sk}_u)$ and, to receive payments, publishes the verification key vk_u (or its hash) as an address. (In fact, there is no limit to the number of addresses that an individual user may possess.)

Transactions. A *transaction* tx represents a payment from a list of input transactions to a list of output recipients. More precisely, tx is specified by a list $\{I_j\}_j$ of inputs and a list $\{O_j\}_j$ of outputs. Each output O_j specifies a value v_j , denominated in *Satoshi* (10^9 Satoshi amounts to 1 bitcoin), and a recipient specification r_j , called **ScriptPubKey**. The specification r_j is given in *Bitcoin script*, a stack-based non-Turing-complete language similar to Forth, and specifies the identity of the recipient of the v_j Satoshi. Each input I_j references an output of a previous transaction tx_j : the reference is specified by a tuple (h_j, k_j, σ_j) , where h_j is the hash of tx_j , k_j is an index specifying which output of tx_j is referenced, and σ_j , called **ScriptSig**, is a an input satisfying the **ScriptPubKey** of the k_j -th output of tx_j . Typically, the **ScriptPubKey** specifies a public key that must sign the transaction spending the output and σ_j contains such a signature, hence their names. Inputs can only be claimed by one transaction to prevent double spending.

The total number of bitcoins output by a transaction, $\sum_j v_j$, cannot exceed the total value of the referenced outputs. Any difference between these two quantities is claimed as a *transaction fee* (see below). Thus, any unspent inputs to a transaction become a fee, and transactions typically have at least two outputs: one to the payment’s recipient and one back to the sender as “change”.

The block chain. Transactions are broadcast in the Bitcoin peer-to-peer network, but are considered valid only once they have been added to the the block chain. To assemble the block chain, *miners* (usually but not necessarily, network nodes) collect transactions from the Bitcoin network and bundle them into *blocks*. Miners then compete for the opportunity to append their own candidate block B to the block chain by searching for a string s such that the integer specified by $\text{SHA256}(\text{SHA256}(B\|s))$ is below some threshold. To incentivize block creation, miners receive a protocol-specified reward (currently 25 BTC) for adding a new block and, moreover, receive per-transaction fees (whose value is specified by the transaction’s creator).

The proof of work protects a block against tampering and also ensures that meaningful computational resources were devoted to finding it. This prevents a sybil attack since all the sybils share the same total computational resources (e.g., the server they are virtualized on). Bitcoin assumes that provided more than half the computational work is held by honest nodes, the block-chain is secure. (Though recent work [ES13] has suggested that the threshold may be larger than 50%.)

A.2 Zerocoin

Zerocoin extends Bitcoin by creating two new transaction types: *mint* and *spend*. A mint transaction allows a user to exchange a quantity of bitcoins for the right to mint a new *zerocoin*. Each zerocoin consists of a digital commitment cm to a random serial number sn . At a later point, a (potentially

different) user may issue a spend transaction containing a destination identity, the serial number sn , and a non-interactive zero-knowledge proof for the NP statement “I know secret cm and r such that (i) cm can be opened to sn with commitment randomness r , and (ii) cm was previously minted at some point in the past”. Crucially, the proof, being zero knowledge, *does not link the spend transaction to any particular mint transaction* (among all mint transactions so far). If the proof verifies correctly and the serial number has not been spent previously, the protocol semantics transfer a corresponding amount of bitcoins to the destination address. In this fashion, Zerocoin functions as a decentralized mix.

Zerocoin uses Pedersen commitments over a prime field \mathbb{F}_p , i.e., $\text{cm} := g^{\text{sn}} h^r$, for random generators g, h of a subgroup of \mathbb{F}_p^* . The corresponding zero-knowledge proofs are constructed by first *accumulating* (via the Strong-RSA accumulator of [CL01]) the set of commitments of all minted zerocoins, and then proving knowledge of the corresponding commitment randomness and membership in this set. For technical reasons, the proof requires a double-discrete-logarithm (DDL) Fiat–Shamir proof of size $\approx |p|\lambda$, where λ is the security parameter. In practice, the size of these proofs exceeds 45 kB at the 128-bit security level, and require 450 ms or more to verify.

Also note that, in Zerocoin, computing the witness for the accumulator requires access to the *entire* set of commitments so far (though the witness can be incrementally updated for each insertion). This technique supports an unlimited number of coins. In contrast, our construction places a cap N on the number of coins (in our implementation, $N = 2^{64}$) but needs only $\log N$ updates to issue N new coins (and these updates can be efficiently batched, cf. Section 8.3.1).

B Completeness of DAP schemes

A DAP scheme $\Pi = (\text{Setup}, \text{CreateAddress}, \text{Mint}, \text{Pour}, \text{VerifyTransaction}, \text{Receive})$ is **complete** if no polynomial-size ledger sampler \mathcal{S} can win the incompleteness experiment with more than negligible probability. In Section 3.4 we informally described this property; we now formally define it.

Definition B.1. Let $\Pi = (\text{Setup}, \text{CreateAddress}, \text{Mint}, \text{Pour}, \text{VerifyTransaction}, \text{Receive})$ be a (candidate) DAP scheme. We say that Π is **complete** if, for every poly(λ)-size ledger sampler \mathcal{S} and sufficiently large λ ,

$$\text{Adv}_{\Pi, \mathcal{S}}^{\text{INCOMP}}(\lambda) < \text{negl}(\lambda) ,$$

where $\text{Adv}_{\Pi, \mathcal{S}}^{\text{INCOMP}}(\lambda) := \Pr[\text{INCOMP}(\Pi, \mathcal{S}, \lambda) = 1]$ is \mathcal{S} ’s advantage in the incompleteness experiment.

We now describe the incompleteness experiment mentioned above. Given a (candidate) DAP scheme Π , a ledger sampler \mathcal{S} , and a security parameter λ , the (probabilistic) experiment $\text{INCOMP}(\Pi, \mathcal{S}, \lambda)$ consists of an interaction between \mathcal{S} and a challenger \mathcal{C} , terminating with a binary output by \mathcal{C} .

At the beginning of the experiment, \mathcal{C} samples $\text{pp} \leftarrow \text{Setup}(1^\lambda)$ and sends pp to \mathcal{S} . Then, \mathcal{S} sends \mathcal{C} a ledger, two coins to be spent, and parameters for a pour transaction; more precisely, \mathcal{S} sends (1) a ledger L ; (2) two coins $\mathbf{c}_1^{\text{old}}, \mathbf{c}_2^{\text{old}}$; (3) two address secret keys $\text{addr}_{\text{sk},1}^{\text{old}}, \text{addr}_{\text{sk},2}^{\text{old}}$; (4) two values $v_1^{\text{new}}, v_2^{\text{new}}$; (5) new address key pairs $(\text{addr}_{\text{pk},1}^{\text{new}}, \text{addr}_{\text{sk},1}^{\text{new}}), (\text{addr}_{\text{pk},2}^{\text{new}}, \text{addr}_{\text{sk},2}^{\text{new}})$; (6) a public value v_{pub} ; and (7) a transaction string info . Afterwards, \mathcal{C} performs various checks on \mathcal{S} ’s message.

Concretely, \mathcal{C} first checks that $\mathbf{c}_1^{\text{old}}$ and $\mathbf{c}_2^{\text{old}}$ are valid unspent coins, i.e., checks that: (i) $\mathbf{c}_1^{\text{old}}$ and $\mathbf{c}_2^{\text{old}}$ are well-formed; (ii) their coin commitments cm_1^{old} and cm_2^{old} appear in (valid) transactions on L ; (iii) their serial numbers sn_1^{old} and sn_2^{old} do *not* appear in (valid) transactions on L . Next, \mathcal{C} checks that $v_1^{\text{new}} + v_2^{\text{new}} + v_{\text{pub}} = v_1^{\text{old}} + v_2^{\text{old}}$ (i.e., the values suggested by \mathcal{S} preserve balance) and $v_1^{\text{old}} + v_2^{\text{old}} \leq v_{\text{max}}$ (i.e., the maximum value is not exceeded). If any of these checks fail, \mathcal{C} aborts and outputs 0.

Otherwise, \mathcal{C} computes rt , the Merkle-tree root over all coin commitments in L (appearing in valid transactions), and, for $i \in \{1, 2\}$, path_i , the authentication path from commitment cm_i^{old} to the root rt . Then, \mathcal{C} attempts to spend $\mathbf{c}_1^{\text{old}}, \mathbf{c}_2^{\text{old}}$ as instructed by \mathcal{S} :

$$(\mathbf{c}_1^{\text{new}}, \mathbf{c}_2^{\text{new}}, \text{tx}_{\text{Pour}}) \leftarrow \text{Pour}(\text{pp}, \text{rt}, \mathbf{c}_1^{\text{old}}, \mathbf{c}_2^{\text{old}}, \text{addr}_{\text{sk},1}^{\text{old}}, \text{addr}_{\text{sk},2}^{\text{old}}, \text{path}_1, \text{path}_2, v_1^{\text{new}}, v_2^{\text{new}}, \text{addr}_{\text{pk},1}^{\text{new}}, \text{addr}_{\text{pk},2}^{\text{new}}, v_{\text{pub}}, \text{info}) .$$

Finally, \mathcal{C} outputs 1 if and only if any of the following conditions hold:

- $\text{tx}_{\text{Pour}} \neq (\text{rt}, \text{sn}_1^{\text{old}}, \text{sn}_2^{\text{old}}, \text{cm}_1^{\text{new}}, \text{cm}_2^{\text{new}}, v_{\text{pub}}, \text{info}, *)$, where $\text{cm}_1^{\text{new}}, \text{cm}_2^{\text{new}}$ are the coin commitments of $\mathbf{c}_1^{\text{new}}, \mathbf{c}_2^{\text{new}}$; OR
- tx_{Pour} is not valid, i.e., $\text{VerifyTransaction}(\text{pp}, \text{tx}_{\text{Pour}}, L)$ outputs 0; OR
- for some $i \in \{1, 2\}$, the coin $\mathbf{c}_i^{\text{new}}$ is not returned by $\text{Receive}(\text{pp}, (\text{addr}_{\text{pk},i}^{\text{new}}, \text{addr}_{\text{sk},i}^{\text{new}}), L')$, where L' is the ledger obtained by appending tx_{Pour} to L .

Remark. There is no need for the challenger \mathcal{C} check that, in turn, both $\mathbf{c}_1^{\text{new}}$ and $\mathbf{c}_2^{\text{new}}$ are spendable, because this follows by induction. Namely, if $\mathbf{c}_1^{\text{new}}, \mathbf{c}_2^{\text{new}}$ were not spendable, a different sampler \mathcal{S}' (that simulates \mathcal{S} and then computes and outputs $\mathbf{c}_1^{\text{new}}$ and $\mathbf{c}_2^{\text{new}}$) would provide a counterexample to the above definition.

C Security of DAP schemes

A DAP scheme $\Pi = (\text{Setup}, \text{CreateAddress}, \text{Mint}, \text{Pour}, \text{VerifyTransaction}, \text{Receive})$ is *secure* if it satisfies ledger indistinguishability, transaction non-malleability, and balance. (See Definition 3.2.) In Section 3.4 we informally described these three properties; we now formally define them.

Each of the definitions employs an experiment involving a (stateful) *DAP oracle* \mathcal{O}^{DAP} that receives and answers queries from an adversary \mathcal{A} (proxied via a challenger \mathcal{C} , which performs the experiment-specific sanity checks). Below, we first describe how \mathcal{O}^{DAP} works.

The oracle \mathcal{O}^{DAP} is initialized by a list of public parameters pp and maintains state. Internally, \mathcal{O}^{DAP} stores: (i) L , a ledger; (ii) ADDR , a set of address key pairs; (iii) COIN , a set of coins. All of $L, \text{ADDR}, \text{COIN}$ start out empty. The oracle \mathcal{O}^{DAP} accepts different types of queries, and each query causes different updates to $L, \text{ADDR}, \text{COIN}$ and outputs. We now describe each type of query Q .

- $Q = (\text{CreateAddress})$
 1. Compute $(\text{addr}_{\text{pk}}, \text{addr}_{\text{sk}}) := \text{CreateAddress}(\text{pp})$.
 2. Add the address key pair $(\text{addr}_{\text{pk}}, \text{addr}_{\text{sk}})$ to ADDR .
 3. Output the address public key addr_{pk} .

The ledger L and coin set COIN remain unchanged.

- $Q = (\text{Mint}, v, \text{addr}_{\text{pk}})$
 1. Compute $(\mathbf{c}, \text{tx}_{\text{Mint}}) := \text{Mint}(\text{pp}, v, \text{addr}_{\text{pk}})$.
 2. Add the coin \mathbf{c} to COIN .
 3. Add the mint transaction tx_{Mint} to L .
 4. Output \perp .

The address set ADDR remains unchanged.

- $Q = (\text{Pour}, \text{idx}_1^{\text{old}}, \text{idx}_2^{\text{old}}, \text{addr}_{\text{pk},1}^{\text{old}}, \text{addr}_{\text{pk},2}^{\text{old}}, \text{info}, v_1^{\text{new}}, v_2^{\text{new}}, \text{addr}_{\text{pk},1}^{\text{new}}, \text{addr}_{\text{pk},2}^{\text{new}}, v_{\text{pub}})$
 1. Compute rt , the root of a Merkle tree over all coin commitments in L .
 2. For each $i \in \{1, 2\}$:
 - (a) Let cm_i^{old} be the $\text{idx}_i^{\text{old}}$ -th coin commitment in L .

- (b) Let tx_i be the mint/pour transaction in L that contains cm_i^{old} .
- (c) Let $\mathbf{c}_i^{\text{old}}$ be the first coin in COIN with coin commitment cm_i^{old} .
- (d) Let $(\text{addr}_{\text{pk},i}^{\text{old}}, \text{addr}_{\text{sk},i}^{\text{old}})$ be the first key pair in ADDR with $\text{addr}_{\text{pk},i}^{\text{old}}$ being $\mathbf{c}_i^{\text{old}}$'s address.
- (e) Compute path_i , the authentication path from cm_i^{old} to rt .
- 3. Compute $(\mathbf{c}_1^{\text{new}}, \mathbf{c}_2^{\text{new}}, \text{tx}_{\text{Pour}}) := \text{Pour}(\text{pp}, \text{rt}, \mathbf{c}_1^{\text{old}}, \mathbf{c}_2^{\text{old}}, \text{addr}_{\text{sk},1}^{\text{old}}, \text{addr}_{\text{sk},2}^{\text{old}}, \text{path}_1, \text{path}_2, v_1^{\text{new}}, v_2^{\text{new}}, \text{addr}_{\text{pk},1}^{\text{new}}, \text{addr}_{\text{pk},2}^{\text{new}}, v_{\text{pub}}, \text{info})$.
- 4. Verify that $\text{VerifyTransaction}(\text{pp}, \text{tx}_{\text{Pour}}, L)$ outputs 1.
- 5. Add the coin $\mathbf{c}_1^{\text{new}}$ to COIN.
- 6. Add the coin $\mathbf{c}_2^{\text{new}}$ to COIN.
- 7. Add the pour transaction tx_{Pour} to L .
- 8. Output \perp .

If any of the above operations fail, the output is \perp (and L , ADDR, COIN remain unchanged).

- $Q = (\text{Receive}, \text{addr}_{\text{pk}})$

1. Look up $(\text{addr}_{\text{pk}}, \text{addr}_{\text{sk}})$ in ADDR. (If no such key pair is found, abort.)
2. Compute $(\mathbf{c}_1, \dots, \mathbf{c}_n) \leftarrow \text{Receive}(\text{pp}, (\text{addr}_{\text{pk}}, \text{addr}_{\text{sk}}), L)$.
3. Add $\mathbf{c}_1, \dots, \mathbf{c}_n$ to COIN.
4. Output $(\text{cm}_1, \dots, \text{cm}_n)$, the corresponding coin commitments.

The ledger L and address set ADDR remain unchanged.

- $Q = (\text{Insert}, \text{tx})$

1. Verify that $\text{VerifyTransaction}(\text{pp}, \text{tx}, L)$ outputs 1. (Else, abort.)
2. Add the mint/pour transaction tx to L .
3. Run **Receive** for all addresses addr_{pk} in ADDR; this updates the COIN with any coins that might have been sent to honest parties via tx .
4. Output \perp .

The address set ADDR remains unchanged.

Remark. The oracle \mathcal{O}^{DAP} provides \mathcal{A} with two ways to cause a pour transaction to be added to L . If \mathcal{A} has already obtained address public keys $\text{addr}_{\text{pk},1}$ and $\text{addr}_{\text{pk},2}$ (via previous **CreateAddress** queries), then \mathcal{A} can use a **Pour** query to elicit a pour transaction tx_{Pour} (despite not knowing address secret keys $\text{addr}_{\text{sk},1}, \text{addr}_{\text{sk},2}$ corresponding to $\text{addr}_{\text{pk},1}, \text{addr}_{\text{pk},2}$). Alternatively, if \mathcal{A} has himself generated both address public keys, then \mathcal{A} knows corresponding address secret keys, and can invoke **Pour** “in his head” to obtain a pour transaction tx_{Pour} , which he can add to L by using an **Insert** query. In the first case, both addresses belong to honest users; in the second, both to \mathcal{A} .

But what about pour transactions where one address belongs to an honest user and one to \mathcal{A} ? Such pour transactions might arise from MPC computations (e.g., to make matching donations). The ledger oracle \mathcal{O}^{DAP} , as defined above, does not support such queries. While extending the definition is straightforward, for simplicity we leave handling such queries to future work.

C.1 Ledger indistinguishability

Ledger indistinguishability is characterized by an experiment **L-IND**, which involves a polynomial-size adversary \mathcal{A} attempting to break a given (candidate) DAP scheme.

Definition C.1. Let $\Pi = (\text{Setup}, \text{CreateAddress}, \text{Mint}, \text{Pour}, \text{VerifyTransaction}, \text{Receive})$ be a (candidate) DAP scheme. We say that Π is **L-IND secure** if, for every $\text{poly}(\lambda)$ -size adversary \mathcal{A} and sufficiently large λ ,

$$\text{Adv}_{\Pi, \mathcal{A}}^{\text{L-IND}}(\lambda) < \text{negl}(\lambda) ,$$

where $\text{Adv}_{\Pi, \mathcal{A}}^{\text{L-IND}}(\lambda) := 2 \cdot \Pr[\text{L-IND}(\Pi, \mathcal{A}, \lambda) = 1] - 1$ is \mathcal{A} 's advantage in the L-IND experiment.

We now describe the L-IND experiment mentioned above. Given a (candidate) DAP scheme Π , adversary \mathcal{A} , and security parameter λ , the (probabilistic) experiment $\text{L-IND}(\Pi, \mathcal{A}, \lambda)$ consists of an interaction between \mathcal{A} and a challenger \mathcal{C} , terminating with a binary output by \mathcal{C} .

At the beginning of the experiment, \mathcal{C} samples $b \in \{0, 1\}$ at random, samples $\text{pp} \leftarrow \text{Setup}(1^\lambda)$, and sends pp to \mathcal{A} ; next, \mathcal{C} initializes (using pp) two separate DAP oracles $\mathcal{O}_0^{\text{DAP}}$ and $\mathcal{O}_1^{\text{DAP}}$ (i.e., the two oracles have separate ledgers and internal tables).

The experiment proceeds in steps and, at each step, \mathcal{C} provides to \mathcal{A} two ledgers $(L_{\text{Left}}, L_{\text{Right}})$, where $L_{\text{Left}} := L_b$ is the current ledger in $\mathcal{O}_b^{\text{DAP}}$ and $L_{\text{Right}} := L_{1-b}$ the one in $\mathcal{O}_{1-b}^{\text{DAP}}$; then \mathcal{A} sends to \mathcal{C} a pair of queries (Q, Q') , which must be of the *same* type (i.e., one of **CreateAddress**, **Mint**, **Pour**, **Receive**, **Insert**). The challenger \mathcal{C} acts differently depending on the query type, as follows.

- If the query type is **Insert**, \mathcal{C} forwards Q to $\mathcal{O}_b^{\text{DAP}}$, and Q' to $\mathcal{O}_{1-b}^{\text{DAP}}$. This allows \mathcal{A} to insert his own transactions directly in L_{Left} and L_{Right} .
- For any other query type, \mathcal{C} first ensures that Q, Q' are *publicly consistent* (see below) and then forwards Q to $\mathcal{O}_0^{\text{DAP}}$, and Q' to $\mathcal{O}_1^{\text{DAP}}$; letting (a_0, a_1) be the two oracle answers, \mathcal{C} replies to \mathcal{A} with (a_b, a_{1-b}) . This allows \mathcal{A} to elicit behavior from honest users. However note that \mathcal{A} does not know the bit b , and hence the mapping between $(L_{\text{Left}}, L_{\text{Right}})$ and (L_0, L_1) ; in other words, \mathcal{A} does not know if he elicits behavior on (L_0, L_1) or on (L_1, L_0) .

At the end of the experiment, \mathcal{A} sends \mathcal{C} a guess $b' \in \{0, 1\}$. If $b = b'$, \mathcal{C} outputs 1; else, \mathcal{C} outputs 0.

Public consistency. As mentioned above, \mathcal{A} sends \mathcal{C} pairs of queries (Q, Q') , which must be of the same type and publicly consistent, a property that we now define. If Q, Q' are both of type **CreateAddress** or **Receive**, then they are always publicly consistent. In the special case of **CreateAddress** we require that both oracles generate the same address. If they are both of type **Mint**, then the minted value in Q must equal that in Q' . Finally, if they are both of type **Pour**, the following restrictions apply.

First, Q, Q' must be individually well-formed; namely, (i) the coin commitments referenced by Q (via the two indices $\text{idx}_1^{\text{old}}, \text{idx}_2^{\text{old}}$) must correspond to coins $\mathbf{c}_1^{\text{old}}, \mathbf{c}_2^{\text{old}}$ that appear in the ledger oracle's coin table **COIN**; (ii) the two coins $\mathbf{c}_1^{\text{old}}, \mathbf{c}_2^{\text{old}}$ must be unspent (i.e. their serial numbers must not appear in a valid pour transactions on the corresponding oracle's ledger); (iii) the address public keys specified in Q must match those in $\mathbf{c}_1^{\text{old}}, \mathbf{c}_2^{\text{old}}$; and (iv) the balance equation must hold (i.e., $v_1^{\text{new}} + v_2^{\text{new}} + v_{\text{pub}} = v_1^{\text{old}} + v_2^{\text{old}}$).

Furthermore, Q, Q' must be jointly consistent with respect to public information and \mathcal{A} 's view; namely: (i) the public values in Q and Q' must equal; (ii) the transaction strings in Q and Q' must equal; (iii) for each $i \in \{1, 2\}$, if the i -th recipient addresses in Q is not in **ADDR** (i.e., belongs to \mathcal{A}) then v_i^{new} in *both* Q and Q' must equal (and vice versa for Q'); and (iv) for each $i \in \{1, 2\}$, if the i -th index in Q references (in L_0) a coin commitment contained in a transaction that was posted via an **Insert** query, then the corresponding index in Q' must reference (in L_1) a coin commitment that also appears in a transaction posted via an **Insert** query and, moreover, v_i^{old} in *both* Q and Q' must equal (and vice versa for Q'). The challenger \mathcal{C} learns v_i^{old} by looking-up the corresponding coin $\mathbf{c}_i^{\text{old}}$ in the oracle's coin set **COIN**. (v) for each $i \in \{1, 2\}$ if the i -th index in Q must not reference a coin that has previously been spent.

C.2 Transaction non-malleability

Transaction non-malleability is characterized by an experiment **TR-NM**, which involves a polynomial-size adversary \mathcal{A} attempting to break a given (candidate) DAP scheme.

Definition C.2. Let $\Pi = (\text{Setup}, \text{CreateAddress}, \text{Mint}, \text{Pour}, \text{VerifyTransaction}, \text{Receive})$ be a (candidate) DAP scheme. We say that Π is **TR-NM secure** if, for every $\text{poly}(\lambda)$ -size adversary \mathcal{A} and sufficiently large λ ,

$$\text{Adv}_{\Pi, \mathcal{A}}^{\text{TR-NM}}(\lambda) < \text{negl}(\lambda) ,$$

where $\text{Adv}_{\Pi, \mathcal{A}}^{\text{TR-NM}}(\lambda) := \Pr[\text{TR-NM}(\Pi, \mathcal{A}, \lambda) = 1]$ is \mathcal{A} 's advantage in the TR-NM experiment.

We now describe the TR-NM experiment mentioned above. Given a (candidate) DAP scheme Π , adversary \mathcal{A} , and security parameter λ , the (probabilistic) experiment $\text{TR-NM}(\Pi, \mathcal{A}, \lambda)$ consists of an interaction between \mathcal{A} and a challenger \mathcal{C} , terminating with a binary output by \mathcal{C} .

At the beginning of the experiment, \mathcal{C} samples $\text{pp} \leftarrow \text{Setup}(1^\lambda)$ and sends pp to \mathcal{A} ; next, \mathcal{C} initializes a DAP oracle \mathcal{O}^{DAP} with pp and allows \mathcal{A} to issue queries to \mathcal{O}^{DAP} . At the end of the experiment, \mathcal{A} sends \mathcal{C} a pour transaction tx^* , and \mathcal{C} outputs 1 if and only if the following conditions hold. Letting \mathcal{T} be the set of pour transactions generated by \mathcal{O}^{DAP} in response to **Pour** queries, there exists $\text{tx} \in \mathcal{T}$ such that: (i) $\text{tx}^* \neq \text{tx}$; (ii) $\text{VerifyTransaction}(\text{pp}, \text{tx}^*, L') = 1$, where L' is the portion of the ledger preceding tx ,²⁷ and (iii) a serial number revealed in tx^* is also revealed in tx .

C.3 Balance

Balance is characterized by an experiment **BAL**, which involves a polynomial-size adversary \mathcal{A} attempting to break a given (candidate) DAP scheme.

Definition C.3. Let $\Pi = (\text{Setup}, \text{CreateAddress}, \text{Mint}, \text{Pour}, \text{VerifyTransaction}, \text{Receive})$ be a (candidate) DAP scheme. We say that Π is **BAL secure** if, for every $\text{poly}(\lambda)$ -size adversary \mathcal{A} and sufficiently large λ ,

$$\text{Adv}_{\Pi, \mathcal{A}}^{\text{BAL}}(\lambda) < \text{negl}(\lambda) ,$$

where $\text{Adv}_{\Pi, \mathcal{A}}^{\text{BAL}}(\lambda) := \Pr[\text{BAL}(\Pi, \mathcal{A}, \lambda) = 1]$ is \mathcal{A} 's advantage in the BAL experiment.

We now describe the BAL experiment mentioned above. Given a (candidate) DAP scheme Π , adversary \mathcal{A} , and security parameter λ , the (probabilistic) experiment $\text{BAL}(\Pi, \mathcal{A}, \lambda)$ consists of an interaction between \mathcal{A} and a challenger \mathcal{C} , terminating with a binary output by \mathcal{C} .

At the beginning of the experiment, \mathcal{C} samples $\text{pp} \leftarrow \text{Setup}(1^\lambda)$, and sends pp to \mathcal{A} ; next, \mathcal{C} (using pp) initializes a DAP oracle \mathcal{O}^{DAP} and allows \mathcal{A} to issue queries to \mathcal{O}^{DAP} . At the conclusion of the experiment, \mathcal{A} sends \mathcal{C} a set of coins S_{coin} . Recalling that **ADDR** is the set of addresses returned by **CreateAddress** queries (i.e., addresses of “honest” users), \mathcal{C} computes the following five quantities.

- v_{Unspent} , the total value of all spendable coins in S_{coin} . The challenger \mathcal{C} can check if a coin $\mathbf{c} \in S_{\text{coin}}$ is spendable as follows: mint a fresh coin \mathbf{c}' of value 0 (via a **Mint** query) and check if a corresponding **Pour** query consuming \mathbf{c}, \mathbf{c}' yields a pour transaction tx_{Pour} that is valid.
- v_{Mint} , the total value of all coins minted by \mathcal{A} . To compute v_{Mint} , the challenger \mathcal{C} sums up the values of all coins that (i) were minted via **Mint** queries using addresses not in **ADDR**, or (ii) whose mint transactions were directly placed on the ledger via **Insert** queries.
- $v_{\text{ADDR} \rightarrow \mathcal{A}}$, the total value payments received by \mathcal{A} from addresses in **ADDR**. To compute $v_{\text{ADDR} \rightarrow \mathcal{A}}$, the challenger \mathcal{C} looks up all pour transactions placed on the ledger via **Pour** queries and sums up the values that were transferred to addresses not in **ADDR**.

²⁷That is, L' is the longest ledger prefix that can be used to spend at least one of the coins spent in tx .

- $v_{\mathcal{A} \rightarrow \text{ADDR}}$, the total value of payments sent by \mathcal{A} to addresses in ADDR . To compute $v_{\mathcal{A} \rightarrow \text{ADDR}}$, the challenger \mathcal{C} first deduces the set $S' \subseteq \text{COIN}$ of all coins received by honest parties and then sums up the values of coins in S' . (Note that \mathcal{C} can compute S' by selecting all coins in COIN that are both tied to an address in ADDR and arose from transactions placed on the ledger by **Insert** queries.)
- v_{Basecoin} , the total value of public outputs placed by \mathcal{A} on the ledger. To compute v_{Basecoin} , the challenger \mathcal{C} looks up all pour transactions placed on the ledger by **Insert** and sums up the corresponding v_{pub} values.

At the end of the experiment, \mathcal{C} outputs 1 if $v_{\text{Unspent}} + v_{\text{Basecoin}} + v_{\mathcal{A} \rightarrow \text{ADDR}} > v_{\text{Mint}} + v_{\text{ADDR} \rightarrow \mathcal{A}}$; else, \mathcal{C} outputs 0.

Remark. There are two methods for \mathcal{A} to spend more public-output money than he owns: (i) by directly inserting transactions on the ledger, and (ii) by asking honest parties to create such transactions. The first method is accounted for in the computation of v_{Basecoin} , while the second method is accounted for in the computation of $v_{\mathcal{A} \rightarrow \text{ADDR}}$ (since \mathcal{A} must first pay the honest party).

D Proof of Theorem 4.1

We prove Theorem 4.1. We omit a formal proof of the completeness claim; one can verify that the DAP scheme's completeness follows, in a straightforward way, from the completeness of the construction's building blocks. Next, we argue security via three separate proofs, respectively showing that our construction satisfies (i) ledger indistinguishability, (ii) transaction non-malleability, and (iii) balance.

D.1 Proof of ledger indistinguishability

We describe a simulation \mathcal{D}_{sim} in which the adversary \mathcal{A} interacts with a challenger \mathcal{C} , as in the L-IND experiment. However \mathcal{D}_{sim} differs from the L-IND experiment in a critical way: all answers sent to \mathcal{A} are computed *independently* of the bit b , so that \mathcal{A} 's advantage in \mathcal{D}_{sim} is 0. The remainder of the proof is devoted to showing that $\text{Adv}_{\Pi, \mathcal{A}}^{\text{L-IND}}(\lambda)$ (i.e., \mathcal{A} 's advantage in the L-IND experiment) is at most negligibly different than \mathcal{A} 's advantage in \mathcal{D}_{sim} .

The simulation. The simulation \mathcal{D}_{sim} works as follows. First, after sampling $b \in \{0, 1\}$ at random, \mathcal{C} samples $\text{pp} \leftarrow \text{Setup}(1^\lambda)$, with the following modification: the zk-SNARK keys are generated as $(\text{pk}_{\text{POUR}}, \text{vk}_{\text{POUR}}, \text{trap}) \leftarrow \text{Sim}(1^\lambda, C_{\text{POUR}})$, to obtain the zero-knowledge trapdoor trap . Then, as in the L-IND experiment, \mathcal{C} sends pp to \mathcal{A} , and then initializes two separate DAP oracles $\mathcal{O}_0^{\text{DAP}}$ and $\mathcal{O}_1^{\text{DAP}}$.

Afterwards, as in L-IND, \mathcal{D}_{sim} proceeds in steps and, at each step, \mathcal{C} provides to \mathcal{A} two ledgers $(L_{\text{Left}}, L_{\text{Right}})$, where $L_{\text{Left}} := L_b$ is the current ledger in $\mathcal{O}_b^{\text{DAP}}$ and $L_{\text{Right}} := L_{1-b}$ the one in $\mathcal{O}_{1-b}^{\text{DAP}}$; then \mathcal{A} sends to \mathcal{C} a message (Q, Q') , which consist of two (publicly-consistent) queries of the same type. The challenger \mathcal{C} acts differently depending on the query type, as follows.

- *Answering CreateAddress queries.* In this case, $Q = Q' = \text{CreateAddress}$.

To answer Q , \mathcal{C} behaves as in L-IND, except for the following modification: after obtaining $(\text{addr}_{\text{pk}}, \text{addr}_{\text{sk}}) \leftarrow \text{CreateAddress}(\text{pp})$, \mathcal{C} replaces a_{pk} in addr_{pk} with a random string of the appropriate length; then, \mathcal{C} stores addr_{sk} in a table and returns addr_{pk} to \mathcal{A} .

Afterwards, \mathcal{C} does the same for Q' .

- *Answering Mint queries.* In this case, $Q = (\mathbf{Mint}, v, \text{addr}_{\text{pk}})$ and $Q' = (\mathbf{Mint}, v, \text{addr}'_{\text{pk}})$.

To answer Q , \mathcal{C} behaves as in L-IND, except for the following modification: the Mint algorithm computes the commitment k as $\text{COMM}_r(\tau \parallel \rho)$, for a random string τ of the appropriate length, instead of as $\text{COMM}_r(a_{\text{pk}} \parallel \rho)$, where a_{pk} is the value specified in addr_{pk} .

Afterwards, \mathcal{C} does the same for Q' .

- *Answering Pour queries.* In this case, Q and Q' both have the form $(\mathbf{Pour}, \text{cm}_1^{\text{old}}, \text{cm}_2^{\text{old}}, \text{addr}_{\text{pk},1}^{\text{old}}, \text{addr}_{\text{pk},2}^{\text{old}}, \text{info}, v_1^{\text{new}}, v_2^{\text{new}}, \text{addr}_{\text{pk},1}^{\text{new}}, \text{addr}_{\text{pk},2}^{\text{new}}, v_{\text{pub}}^{\text{new}})$.

To answer Q , \mathcal{C} modifies the way some values are computed:

1. Compute rt_i by accumulating all of the valid coin commitments on L_i .
2. Set v_{pub} and info to the corresponding input values.
3. For each $j \in \{1, 2\}$:
 - (a) Sample a uniformly random sn_j^{old} .
 - (b) If $\text{addr}_{\text{pk},j}^{\text{new}}$ is an address generated by a previous query to **CreateAddress**, (i) sample a coin commitment cm_j^{new} on a random input, (ii) run $\mathcal{K}_{\text{enc}}(\text{pp}_{\text{enc}}) \rightarrow (\text{pk}_{\text{enc}}, \text{sk}_{\text{enc}})$ and compute $\mathbf{C}_j^{\text{new}} := \mathcal{E}_{\text{enc}}(\text{pk}_{\text{enc}}, r)$ for a random r of suitable length.
 - (c) Otherwise, calculate $(\text{cm}_j^{\text{new}}, \mathbf{C}_j^{\text{new}})$ as in the Pour algorithm.²⁸
4. Set h_1 and h_2 to be random strings of the appropriate length.
5. Compute all remaining values as in the Pour algorithm
6. The pour proof is computed as $\pi_{\text{POUR}} := \text{Sim}(\text{trap}, x)$, where $x := (\text{rt}, \text{sn}_1^{\text{old}}, \text{sn}_2^{\text{old}}, \text{cm}_1^{\text{new}}, \text{cm}_2^{\text{new}}, v_{\text{pub}}, h_1, h_2)$.

Afterwards, \mathcal{C} does the same for Q' .

- *Answering Receive queries.* In this case, $Q = (\mathbf{Receive}, \text{addr}_{\text{pk}})$ and $Q' = (\mathbf{Receive}, \text{addr}'_{\text{pk}})$. The answer to each query proceeds as in the L-IND experiment.

- *Answering Insert queries.* In this case, $Q = (\mathbf{Insert}, \text{tx})$ and $Q' = (\mathbf{Insert}, \text{tx}')$. The answer to each query proceeds as in the L-IND experiment.

In each of the above cases, the response to \mathcal{A} is computed independently of the bit b . Thus, when \mathcal{A} outputs a guess b' , it must be the case that $\Pr[b = b'] = 1/2$, i.e., \mathcal{A} 's advantage in \mathcal{D}_{sim} is 0.

Proof that the simulation is indistinguishable from the real experiment. We now describe a sequence of hybrid experiments $(\mathcal{D}_{\text{real}}, \mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_{\text{sim}})$ in each of which a challenger \mathcal{C} conducts a modification of the L-IND experiment with \mathcal{A} . We define $\mathcal{D}_{\text{real}}$ to be the original L-IND experiment, and \mathcal{D}_{sim} to be the simulation described above.

With a slight abuse of notation, given experiment \mathcal{D} , we define $\text{Adv}^{\mathcal{D}}$ to be the absolute value of the difference between (i) the L-IND advantage of \mathcal{A} in \mathcal{D} and (ii) the L-IND advantage of \mathcal{A} in $\mathcal{D}_{\text{real}}$. Also, let

- $q_{\mathbf{CA}}$ be the total number of **CreateAddress** queries issued by \mathcal{A} ,
- $q_{\mathbf{P}}$ be the total number of **Pour** queries issued by \mathcal{A} , and
- $q_{\mathbf{M}}$ be the total number of **Mint** queries issued by \mathcal{A} .

Finally, define Adv^{Enc} to be \mathcal{A} 's advantage in Enc's IND-CCA and IK-CCA experiments, Adv^{PRF} to be \mathcal{A} 's advantage in distinguishing the pseudorandom function PRF from a random one, and Adv^{COMM} to be \mathcal{A} 's advantage against the hiding property of COMM.

We now describe each of the hybrid experiments.

²⁸Note that by the restrictions of the experiment, the value v_i^{new} is identical between Q_{Left} and Q_{Right} .

- **Experiment \mathcal{D}_1 .** The experiment \mathcal{D}_1 modifies $\mathcal{D}_{\text{real}}$ by simulating the zk-SNARKs. More precisely, we modify $\mathcal{D}_{\text{real}}$ so that \mathcal{C} simulates each zk-SNARK proof, as follows. At the beginning of the experiment, instead of invoking $\text{KeyGen}(1^\lambda, C_{\text{POUR}})$, \mathcal{C} invokes $\text{Sim}(1^\lambda, C_{\text{POUR}})$ and obtains $(\mathbf{pk}_{\text{POUR}}, \mathbf{vk}_{\text{POUR}}, \mathbf{trap})$. At each subsequent invocation of the **Pour** algorithm, \mathcal{C} computes $\pi_{\text{POUR}} \leftarrow \text{Sim}(\mathbf{trap}, x)$, without using any witnesses, instead of using **Prove**. Since the zk-SNARK system is perfect zero knowledge, the distribution of the simulated π_{POUR} is identical to that of the proofs computed in $\mathcal{D}_{\text{real}}$. Hence $\text{Adv}^{\mathcal{D}_1} = 0$.
- **Experiment \mathcal{D}_2 .** The experiment \mathcal{D}_2 modifies \mathcal{D}_1 by replacing the ciphertexts in a pour transaction by encryptions of random strings. More precisely, we modify \mathcal{D}_1 so that, each time \mathcal{A} issues a **Pour** query where one of the output addresses $(\mathbf{addr}_{\mathbf{pk},1}^{\text{new}}, \mathbf{addr}_{\mathbf{pk},2}^{\text{new}})$ is in the set of addresses previously generated by a **CreateAddress** query, the two ciphertexts $\mathbf{C}_1^{\text{new}}, \mathbf{C}_2^{\text{new}}$ are generated as follows: (i) $(\mathbf{pk}_{\text{enc}}^{\text{new}}, \mathbf{sk}_{\text{enc}}^{\text{new}}) \leftarrow \mathcal{K}_{\text{enc}}(\mathbf{pp}_{\text{enc}})$; (ii) for each $j \in \{1, 2\}$, $\mathbf{C}_j^{\text{new}} := \mathcal{E}_{\text{enc}}(\mathbf{pk}_{\text{enc},j}^{\text{new}}, r)$ where r is a message sampled uniformly from the plaintext space of the encryption scheme. By Lemma D.1 (see below), $|\text{Adv}^{\mathcal{D}_2} - \text{Adv}^{\mathcal{D}_1}| \leq 4 \cdot q_{\mathbf{P}} \cdot \text{Adv}^{\text{Enc}}$.
- **Experiment \mathcal{D}_3 .** The experiment \mathcal{D}_3 modifies \mathcal{D}_2 by replacing all PRF-generated values with random strings. More precisely, we modify \mathcal{D}_2 so that:
 - each time \mathcal{A} issues a **CreateAddress** query, the value $a_{\mathbf{pk}}$ within the returned $\mathbf{addr}_{\mathbf{pk}}$ is substituted with a random string of the same length;
 - each time \mathcal{A} issues a **Pour** query, each of the serial numbers $\mathbf{sn}_1^{\text{old}}, \mathbf{sn}_2^{\text{old}}$ in $\mathbf{tx}_{\text{Pour}}$ is substituted with a random string of the same length, and h_{info} with a random string of the same length.
By Lemma D.2 (see below), $|\text{Adv}^{\mathcal{D}_3} - \text{Adv}^{\mathcal{D}_2}| \leq q_{\mathbf{CA}} \cdot \text{Adv}^{\text{PRF}}$.
- **Experiment \mathcal{D}_{sim} .** The experiment \mathcal{D}_{sim} is already described above. For comparison, we explain how it differs from \mathcal{D}_3 : the coin commitments are replaced with commitments to random inputs. More precisely, we modify \mathcal{D}_3 so that:
 - each time \mathcal{A} issues a **Mint** query, the coin commitment \mathbf{cm} in $\mathbf{tx}_{\text{Mint}}$ is substituted with a commitment to a random input; and
 - each time \mathcal{A} issues a **Pour** query, then, for each $j \in \{1, 2\}$, if the output address $\mathbf{addr}_{\mathbf{pk},j}^{\text{new}}$ is in the set of addresses previously generated by an **CreateAddress** query, $\mathbf{cm}_j^{\text{new}}$ is substituted with a commitment to a random input.
By Lemma D.3 (see below), $|\text{Adv}^{\mathcal{D}_{\text{sim}}} - \text{Adv}^{\mathcal{D}_3}| \leq (q_{\mathbf{M}} + 4 \cdot q_{\mathbf{P}}) \cdot \text{Adv}^{\text{COMM}}$.

As argued above, the responses provided to \mathcal{A} in \mathcal{D}_{sim} are independent of the bit b , so that $\text{Adv}^{\mathcal{D}_{\text{sim}}} = 0$. Then, by summing over \mathcal{A} 's advantages in the hybrid experiments, we can bound \mathcal{A} 's advantage in $\mathcal{D}_{\text{real}}$ by

$$\text{Adv}_{\text{II}, \mathcal{A}}^{\text{L-IND}}(\lambda) \leq 4 \cdot q_{\mathbf{P}} \cdot \text{Adv}^{\text{Enc}} + q_{\mathbf{CA}} \cdot \text{Adv}^{\text{PRF}} + (q_{\mathbf{M}} + 4 \cdot q_{\mathbf{P}}) \cdot \text{Adv}^{\text{COMM}},$$

which is negligible in λ . This concludes the proof of ledger indistinguishability. Below, we sketch proofs for the lemmas used above (Lemma D.1, Lemma D.2, and Lemma D.3).

Lemma D.1. Let Adv^{Enc} be the maximum of:

- \mathcal{A} 's advantage in the IND-CCA experiment against the encryption scheme Enc , and
- \mathcal{A} 's advantage in the IK-CCA experiment against the encryption scheme Enc .

Then after $q_{\mathbf{P}}$ **Pour** queries, $|\text{Adv}^{\mathcal{D}_2} - \text{Adv}^{\mathcal{D}_1}| \leq 4 \cdot q_{\mathbf{P}} \cdot \text{Adv}^{\text{Enc}}$.

Proof sketch. Define $\epsilon := \text{Adv}^{\mathcal{D}_2} - \text{Adv}^{\mathcal{D}_1}$. Using \mathcal{A} , we first show how to construct a solver with advantage $\geq \frac{\epsilon}{2 \cdot q_{\mathbf{P}}}$ in the IK-CCA or IND-CCA experiments. We use a hybrid \mathbf{H} , intermediate between \mathcal{D}_1 and \mathcal{D}_2 ; concretely, \mathbf{H} modifies \mathcal{D}_1 so that each ciphertext (where the corresponding public key appears in the set generated by a **CreateAddress** query) is replaced with the encryption of the same plaintext, but under a new, random public key generated via the \mathcal{K}_{enc} algorithm. (For comparison, \mathcal{D}_2 modifies \mathbf{H} so that each plaintext is replaced with a random plaintext drawn from the plaintext space.) We now argue that \mathcal{A} 's advantage in distinguishing \mathbf{H} and \mathcal{D}_1 is at most $2 \cdot q_{\mathbf{P}} \cdot \text{Adv}^{\text{Enc}}$, and so is for distinguishing \mathcal{D}_2 and \mathbf{H} . Overall, we deduce that $|\text{Adv}^{\mathcal{D}_2} - \text{Adv}^{\mathcal{D}_1}| \leq 4 \cdot q_{\mathbf{P}} \cdot \text{Adv}^{\text{Enc}}$.

First, we discuss \mathbf{H} and \mathcal{D}_1 . For some $j \in \{1, \dots, q_{\mathbf{CA}}\}$, when \mathcal{A} makes the j -th query of the form **CreateAddress**, query the IK-CCA challenger to obtain two public keys $(\mathbf{pk}_{\text{enc},0}, \mathbf{pk}_{\text{enc},1})$ and return $\mathbf{pk}_{\text{enc}} := \mathbf{pk}_{\text{enc},0}$ in the response to \mathcal{A} . At the time \mathcal{A} issues a **Pour** query that results in the i -th ciphertext \mathbf{C}_i being encrypted under \mathbf{pk}_{enc} , query the IK-CCA challenger on the corresponding plaintext m and receive $\mathbf{C}^* = \mathcal{E}_{\text{enc}}(\mathbf{pk}_{\text{enc}}, \bar{b}, m)$ where \bar{b} is the bit chosen by the IK-CCA challenger. Substitute $\mathbf{C}_i := \mathbf{C}^*$ and write the resulting tx_{Pour} to the Ledger. When \mathcal{A} outputs b' we return this guess as our guess in the IK-CCA experiment. We note that when $\bar{b} = 0$ then \mathcal{A} 's view of the interaction is distributed identically to that of \mathcal{D}_1 , and when $\bar{b} = 1$ then \mathcal{A} 's view represents an intermediate hybrid where one key has been substituted. By a standard hybrid argument over each of the $2 \cdot q_{\mathbf{P}}$ ciphertexts, we note that over the random coins of the experiment, our solver must succeed in the IK-CCA experiment with advantage $\geq \frac{\epsilon}{2 \cdot q_{\mathbf{P}}}$. If we assume a maximum adversarial advantage Adv^{Enc} against the IK-CCA experiment for the encryption scheme, then we get that $|\text{Adv}^{\mathbf{H}} - \text{Adv}^{\mathcal{D}_2}| \leq 2 \cdot q_{\mathbf{P}} \cdot \text{Adv}^{\text{Enc}}$.

Next, we discuss \mathcal{D}_2 and \mathbf{H} ; the argument is similar to the above one. This time, rather than replacing the key used to encrypt, we replace the plaintext with a random message drawn from the plaintext space; this final distribution is the same as in \mathcal{D}_2 . We omit the formal description of the resulting IND-CCA solver (which essentially follows the pattern above), and simply note that $|\text{Adv}^{\mathcal{D}_2} - \text{Adv}^{\mathbf{H}}| \leq 2 \cdot q_{\mathbf{P}} \cdot \text{Adv}^{\text{Enc}}$. \square

Lemma D.2. Let Adv^{PRF} be \mathcal{A} 's advantage in distinguishing the pseudorandom function PRF from a random function. Then, after $q_{\mathbf{CA}}$ **CreateAddress** queries, $|\text{Adv}^{\mathcal{D}_3} - \text{Adv}^{\mathcal{D}_2}| \leq q_{\mathbf{CA}} \cdot \text{Adv}^{\text{PRF}}$.

Proof sketch. We first describe a hybrid \mathbf{H} , intermediate between \mathcal{D}_2 and \mathcal{D}_3 , in which all values computed using the first (rather than all) oracle-generated key $a_{\mathbf{sk}}$ are replaced with random strings. Then, we show that \mathcal{A} 's advantage in distinguishing between \mathbf{H} and \mathcal{D}_2 is at most Adv^{PRF} . Finally, we extend the argument to all $q_{\mathbf{CA}}$ oracle-generated keys (corresponding to what happens in \mathcal{D}_3).

We now describe \mathbf{H} . On receiving \mathcal{A} 's first **CreateAddress** query, replace the public address $\text{addr}_{\mathbf{pk}} = (a_{\mathbf{pk}}, \mathbf{pk}_{\text{enc}})$ with $\text{addr}_{\mathbf{pk}} = (\tau, \mathbf{pk}_{\text{enc}})$ where τ is a random string of the appropriate length. On each subsequent **Pour** query, for each $i \in \{1, 2\}$, if $\text{addr}_{\mathbf{pk},i}^{\text{old}} = \text{addr}_{\mathbf{pk}}$ then:

1. in the output tx_{Pour} , replace sn_i^{old} with a random string of appropriate length;
2. in the output tx_{Pour} , replace each of h_1, h_2 with a random string of appropriate length.
3. simulate the zk-SNARK proof π_{POUR} for the new transaction.

Note that the above modifications do not affect the computation of the zk-SNARK proof π_{POUR} , because π_{POUR} is simulated with the help of a trapdoor.

We now argue that \mathcal{A} 's advantage in distinguishing between \mathbf{H} and \mathcal{D}_2 is at most Adv^{PRF} . Let $a_{\mathbf{sk}}$ be the random, secret seed for PRF generated by the oracle in answering the first **CreateAddress** query. In \mathcal{D}_2 (as in $\mathcal{D}_{\text{real}}$):

- $a_{\mathbf{pk}} := \text{PRF}_{a_{\mathbf{sk}}}^{\text{addr}}(0)$;
- for each $i \in \{1, 2\}$, $\text{sn}_i := \text{PRF}_{a_{\mathbf{sk}}}^{\text{sn}}(\rho)$ for a random (and not previously used) ρ

- for each $i \in \{1, 2\}$, $h_i := \text{PRF}_{a_{\text{sk}}}^{\text{pk}}(i \| h_{\text{Sig}})$ and, with overwhelming probability, h_{Sig} is unique.
- Moreover, each of $\text{PRF}_{a_{\text{sk}}}^{\text{addr}}$, $\text{PRF}_{a_{\text{sk}}}^{\text{sn}}$, $\text{PRF}_{a_{\text{sk}}}^{\text{pk}}$ are constructed from $\text{PRF}_{a_{\text{sk}}}$ as specified in Section 4.1. Note that, with overwhelming probability, no two calls to $\text{PRF}_{a_{\text{sk}}}$ are made on the same input. First, even identical inputs passed to $\text{PRF}_{a_{\text{sk}}}^{\text{addr}}$, $\text{PRF}_{a_{\text{sk}}}^{\text{sn}}$, $\text{PRF}_{a_{\text{sk}}}^{\text{pk}}$ produce different underlying calls to $\text{PRF}_{a_{\text{sk}}}$. Second, within each construction, there is exactly one call to $\text{PRF}_{a_{\text{sk}}}^{\text{addr}}$, and the calls to $\text{PRF}_{a_{\text{sk}}}^{\text{sn}}$ are each by definition unique. Finally, with overwhelming probability, the calls to $\text{PRF}_{a_{\text{sk}}}^{\text{pk}}$ from different transactions each reference a distinct digest h_{Sig} , and, within a given transaction, the two calls each begin with a distinct prefix.

Now let \mathcal{O} be an oracle that implements either $\text{PRF}_{a_{\text{sk}}}$ or a random function. We show that if \mathcal{A} distinguishes \mathbf{H} from \mathcal{D}_2 with probability ϵ , then we can construct a distinguisher for the two cases of \mathcal{O} . In either case we use \mathcal{O} to generate all values computed using $\text{PRF}_{a_{\text{sk}}}^{\text{addr}}$, $\text{PRF}_{a_{\text{sk}}}^{\text{sn}}$, $\text{PRF}_{a_{\text{sk}}}^{\text{pk}}$. Clearly, when \mathcal{O} implements $\text{PRF}_{a_{\text{sk}}}$, the distribution of the experiment is identical to \mathcal{D}_2 ; instead, when \mathcal{O} implements a random function, the distribution of the experiment is identical to \mathbf{H} . Thus, \mathcal{A} 's advantage is at most Adv^{PRF} .

Finally, by a standard hybrid argument, we extend the above to all q_{CA} oracle-generated addresses; then, \mathcal{A} 's differential distinguishing advantage is at most $q_{\text{CA}} \cdot \text{Adv}^{\text{PRF}}$. Because this final hybrid is equal to \mathcal{D}_3 , we deduce that $|\text{Adv}^{\mathcal{D}_3} - \text{Adv}^{\mathcal{D}_2}| \leq q_{\text{CA}} \cdot \text{Adv}^{\text{PRF}}$. \square

Lemma D.3. Let Adv^{COMM} be \mathcal{A} 's advantage against the hiding property of COMM . After q_{M} **Mint** queries and q_{P} **Pour** queries, $|\text{Adv}^{\mathcal{D}_{\text{sim}}} - \text{Adv}^{\mathcal{D}_3}| \leq (q_{\text{M}} + 4 \cdot q_{\text{P}}) \cdot \text{Adv}^{\text{COMM}}$.

Proof sketch. We only provide a short sketch, because the structure of the argument is similar to the one used to prove Lemma D.2 above.

For the first **Mint** or **Pour** query, replace the “internal” commitment $k := \text{COMM}_r(a_{\text{pk}} \| \rho)$ with a random string of the appropriate length. Since ρ is random (and unique), then \mathcal{A} 's advantage in distinguishing this modified experiment from \mathcal{D}_2 is at most Adv^{COMM} . Then, if we similarly modify all q_{M} **Mint** queries and all q_{P} **Pour** queries, by replacing the resulting $q_{\text{M}} + 2 \cdot q_{\text{P}}$ internal commitments with random strings, we can bound \mathcal{A} 's advantage by $(q_{\text{M}} + 2 \cdot q_{\text{P}}) \cdot \text{Adv}^{\text{COMM}}$.

Next, in a similar vein, if replace the coin commitment in the first **Pour** with a commitment to a random value, then \mathcal{A} 's advantage in distinguishing this modified experiment from the above one is at most Adv^{COMM} . Then, if we similarly modify all q_{P} **Pour** queries, by replacing the resulting $2 \cdot q_{\text{P}}$ coin commitments with random strings, we obtain the experiment \mathcal{D}_{sim} , and deduce that $|\text{Adv}^{\mathcal{D}_{\text{sim}}} - \text{Adv}^{\mathcal{D}_3}| \leq (q_{\text{M}} + 4 \cdot q_{\text{P}}) \cdot \text{Adv}^{\text{COMM}}$. \square

D.2 Proof of transaction non-malleability

Letting \mathcal{T} be the set of pour transactions generated by \mathcal{O}^{DAP} in response to **Pour** queries, recall that \mathcal{A} wins the TR-NM experiment whenever it outputs tx^* such that there exists $\text{tx}' \in \mathcal{T}$ such that: (i) $\text{tx}^* \neq \text{tx}'$; (ii) $\text{VerifyTransaction}(\text{pp}, \text{tx}^*, L') = 1$, where L' is the portion of the ledger preceding tx' ; and (iii) a serial number revealed in tx^* is also revealed in tx' . Being a pour transaction, tx^* has the form $(\text{rt}, \text{sn}_1^{\text{old}}, \text{sn}_2^{\text{old}}, \text{cm}_1^{\text{new}}, \text{cm}_2^{\text{new}}, v_{\text{pub}}, \text{info}, *)$, where $* := (\text{pk}_{\text{sig}}, h_1, h_2, \pi_{\text{POUR}}, \mathbf{C}_1, \mathbf{C}_2, \sigma)$; set $h_{\text{Sig}} := \text{CRH}(\text{pk}_{\text{sig}})$. Let pk'_{sig} be the corresponding public key in tx' and set $h'_{\text{Sig}} := \text{CRH}(\text{pk}'_{\text{sig}})$.

Define $\epsilon := \text{Adv}_{\Pi, \mathcal{A}}^{\text{TR-NM}}(\lambda)$, and let $\mathcal{Q}_{\text{CA}} = \{a_{\text{sk},1}, \dots, a_{\text{sk},q_{\text{CA}}}\}$ be the set of internal address keys created by \mathcal{C} in response to \mathcal{A} 's **CreateAddress** queries. Let $\mathcal{Q}_{\text{P}} = (\text{pk}_{\text{sig},1}, \dots, \text{pk}_{\text{sig},q_{\text{P}}})$ be the set of signature public keys created by \mathcal{C} in response to \mathcal{A} 's **Pour** queries. We decompose the event in which \mathcal{A} wins into the following four disjoint events.

- **EVENT_{sig}**: \mathcal{A} wins, and there is $\text{pk}_{\text{sig}}'' \in \mathcal{Q}_{\text{P}}$ such that $\text{pk}_{\text{sig}}'' = \text{pk}'_{\text{sig}}$.

- $\text{EVENT}_{\text{col}}$: \mathcal{A} wins, the above event does not occur, and there is $\text{pk}_{\text{sig}}'' \in \mathcal{Q}_{\mathbf{P}}$ such that $h_{\text{Sig}} = \text{CRH}(\text{pk}_{\text{sig}}'')$.
- $\text{EVENT}_{\text{mac}}$: \mathcal{A} wins, the above two events do not occur, and $h_i = \text{PRF}_a^{\text{pk}}(i \| h_{\text{Sig}})$ for some $i \in \{1, 2\}$ and $a \in \mathcal{Q}_{\mathbf{CA}}$.
- $\text{EVENT}_{\text{key}}$: \mathcal{A} wins, the above three events do not occur, and $h_i \neq \text{PRF}_a^{\text{pk}}(i \| h_{\text{Sig}})$ for all $i \in \{1, 2\}$ and $a \in \mathcal{Q}_{\mathbf{CA}}$.

Clearly, $\epsilon = \Pr[\text{EVENT}_{\text{sig}}] + \Pr[\text{EVENT}_{\text{col}}] + \Pr[\text{EVENT}_{\text{key}}] + \Pr[\text{EVENT}_{\text{mac}}]$. Hence, to show that ϵ is negligible in λ , it suffices to argue that each of these probabilities is negligible in λ .

Bounding the probability of Event_{sig}. Define $\epsilon_1 := \Pr[\text{EVENT}_{\text{sig}}]$. Let σ be the signature in tx^* , and σ'' be the signature in the first pour transaction $\text{tx}'' \in \mathcal{T}$ that contains pk_{sig}'' . When $\text{EVENT}_{\text{sig}}$ occurs, since $\text{pk}_{\text{sig}} = \text{pk}_{\text{sig}}''$, the two signatures are with respect to the same public key. Moreover, since tx^* is valid, $\mathcal{V}_{\text{sig}}(\text{pk}_{\text{sig}}, m, \sigma) = 1$ where m is everything in tx^* but for σ . Let m'' consist of all elements in tx'' but for σ'' . Observe that whenever $\text{tx}^* \neq \text{tx}''$ we also have $(m, \sigma) \neq (m'', \sigma'')$. We use this fact below to show that \mathcal{A} forges a signature with non-negligible probability.

First, we argue that, conditioned on $\text{EVENT}_{\text{sig}}$, $\text{tx}^* \neq \text{tx}''$ with overwhelming probability; we do so by way of contradiction. First, since \mathcal{A} wins, by definition there is $\text{tx}' \in \mathcal{T}$ such that $\text{tx}^* \neq \text{tx}'$ and yet each of tx^* and tx' share one serial number. Therefore: (i) $\text{tx}^* \neq \text{tx}'$; and (ii) if $\text{tx}^* = \text{tx}''$ then tx'' and tx' also share a serial number. However the probability that tx' and tx'' share a serial number is bounded by the probability \tilde{p} that \mathcal{T} contains two transactions that share the same serial number. Because each serial number is computed as $\text{PRF}_{a_{\text{sk}}}^{\text{sn}}(\rho)$, where ρ is random, \tilde{p} is negligible. We conclude that $\text{tx}^* \neq \text{tx}''$ with all but negligible probability.

Next, we describe an algorithm \mathcal{B} , which uses \mathcal{A} as a subroutine, that wins the SUF-1CMA game against Sig with probability $\epsilon_1/q_{\mathbf{P}}$. After receiving a verification key pk_{sig}'' from the SUF-1CMA challenger, the algorithm \mathcal{B} performs the following steps.

1. \mathcal{B} selects a random index $j \leftarrow \{1, \dots, q_{\mathbf{P}}\}$.
2. \mathcal{B} conducts the TR-NM experiment with \mathcal{A} , except that, when \mathcal{A} issues the j -th **Pour** query, \mathcal{B} executes Pour as usual, but modifies the resulting pour transaction tx'' as follows: (i) it substitutes pk_{sig}'' for the signature public key in tx'' ; (ii) it queries the SUF-1CMA challenger to obtain σ'' on the appropriate message m'' ; and (iii) it substitutes σ'' for the signature in tx'' .
3. When \mathcal{A} outputs tx^* , \mathcal{B} looks into tx^* to obtain pk_{sig} , m , and σ .
4. If $\text{pk}_{\text{sig}} \neq \text{pk}_{\text{sig}}''$ then \mathcal{B} aborts; otherwise \mathcal{B} outputs (m, σ) as a forgery for Sig .

Note that tx'' has the same distribution as an “untampered” pour transaction; thus, all transactions returned to \mathcal{A} are distributed as in the TR-NM experiment. Since the index j is selected at random, \mathcal{B} succeeds in the experiment with probability at least $\epsilon_1/q_{\mathbf{P}}$. Because Sig is SUF-1CMA, ϵ_1 must be negligible in λ .

Bounding the probability of Event_{col}. Define $\epsilon_2 := \Pr[\text{EVENT}_{\text{col}}]$. When $\text{EVENT}_{\text{col}}$ occurs, \mathcal{A} receives a transaction tx' containing a public key pk_{sig}'' , and subsequently outputs a transaction tx^* containing a public key pk_{sig} such that (i) $\text{pk}_{\text{sig}} \neq \text{pk}_{\text{sig}}''$, but (ii) $\text{CRH}(\text{pk}_{\text{sig}}) = \text{CRH}(\text{pk}_{\text{sig}}'')$. In particular, \mathcal{A} finds collisions for CRH with probability ϵ_2 . Because CRH is collision resistant, ϵ_2 must be negligible in λ .

Bounding the probability of Event_{mac}. Define $\epsilon_3 := \Pr[\text{EVENT}_{\text{mac}}]$. We first define an experiment \mathcal{D}_1 , which modifies the TR-NM experiment as follows. When \mathcal{C} samples $\text{pp} \leftarrow \text{Setup}(1^\lambda)$, the sub-call to $(\text{pk}_{\text{POUR}}, \text{vk}_{\text{POUR}}) \leftarrow \text{KeyGen}(1^\lambda, C_{\text{POUR}})$ is replaced by $(\text{pk}_{\text{POUR}}, \text{vk}_{\text{POUR}}, \text{trap}) \leftarrow \text{Sim}(1^\lambda, C_{\text{POUR}})$, so to obtain the zero-knowledge trapdoor trap . Afterwards, each time \mathcal{A} issues a **Pour** query, \mathcal{C} replaces the zk-SNARK proof in the resulting pour transaction with a simulated proof, obtained by running $\text{Sim}(\text{trap}, x)$ for an appropriate input x . Because the zk-SNARK is perfect zero knowledge, $\Pr[\text{EVENT}_{\text{mac}}] = \epsilon_3$ in the \mathcal{D}_1 experiment as well.

Assume by way of contradiction that ϵ_3 is non-negligible. We now show how to construct an attacker \mathcal{B} , which uses \mathcal{A} as a subroutine, that distinguishes PRF from a random function RAND with non-negligible probability. The algorithm \mathcal{B} , which has access either to $\mathcal{O} = \text{PRF}$ or $\mathcal{O} = \text{RAND}$, “interfaces” between \mathcal{A} and \mathcal{C} in the experiment \mathcal{D}_1 above, as follows.

1. First, \mathcal{B} selects a random index $j \leftarrow \{1, \dots, q_{\mathbf{CA}}\}$, which identifies $a_{\mathbf{sk},j} \in \mathcal{Q}_{\mathbf{CA}}$.
2. Next, \mathcal{B} uses the oracle \mathcal{O} instead of $\text{PRF}_{a_{\mathbf{sk},j}}$, i.e., anytime a value needs to be computed depending on $\text{PRF}_{a_{\mathbf{sk},j}}(z)$, for some z , $\mathcal{O}(z)$ is used instead. (For instance, the public address key $a_{\mathbf{pk},j}$ is one such value.)
3. Finally, after \mathcal{A} outputs tx^* :
 - (a) if \mathcal{O} has been previously evaluated the expression “ $\text{PRF}_{a_{\mathbf{sk},j}}^{\text{pk}}(i \| h_{\text{Sig}})$ ” using \mathcal{O} , \mathcal{B} aborts and outputs 1;
 - (b) otherwise, \mathcal{B} evaluates the expression “ $\text{PRF}_{a_{\mathbf{sk},j}}^{\text{pk}}(i \| h_{\text{Sig}})$ ” by using \mathcal{O} ; if the result equals h_i , \mathcal{B} outputs 1, else it outputs 0.

Conducting the above strategy does not require knowledge of $a_{\mathbf{sk},j}$ because, having the simulation trapdoor, \mathcal{B} does not need witnesses to generate (valid) zk-SNARK proofs.

We now argue that $|\Pr[\mathcal{B}^{\text{PRF}}(1^\lambda) = 1] - \Pr[\mathcal{B}^{\text{RAND}}(1^\lambda) = 1]|$ is non-negligible.

- *Case 1: $\mathcal{O} = \text{RAND}$.* Observe that:

$$\Pr[\mathcal{B}^{\text{RAND}}(1^\lambda) = 1 \mid \mathcal{B}^{\text{RAND}}(1^\lambda) \text{ does not abort}] = 2^{-\omega} .$$

where ω is the output length of PRF. Hence:

$$\Pr[\mathcal{B}^{\text{RAND}}(1^\lambda) = 1] = \left(1 - \Pr[\mathcal{B}^{\text{RAND}}(1^\lambda) \text{ aborts}]\right) \cdot 2^{-\omega} + \Pr[\mathcal{B}^{\text{RAND}}(1^\lambda) \text{ aborts}].$$

- *Case 2: $\mathcal{O} = \text{PRF}$.* In this case the distribution of the simulation is identical to that of \mathcal{D}_1 , and \mathcal{B} has set $a_{\mathbf{sk},j}$ equal to the seed used by \mathcal{O} . Recall that, when $\text{EVENT}_{\text{mac}}$ holds, $h_i = \text{PRF}_a^{\text{pk}}(i \| h_{\text{Sig}})$ for some $a \in \mathcal{Q}_{\mathbf{CA}}$. Since \mathcal{A} ’s view of the experiment is independent of j , the probability that $a = a_{\mathbf{sk},j}$ is at least $1/q_{\mathbf{CA}}$, and the probability that $h_i = \text{PRF}_{a_{\mathbf{sk},j}}^{\text{pk}}(i \| h_{\text{Sig}})$ is at least $\epsilon_3/q_{\mathbf{CA}}$. Hence:

$$\Pr[\mathcal{B}^{\text{PRF}}(1^\lambda) = 1 \mid \mathcal{B}^{\text{PRF}}(1^\lambda) \text{ does not abort}] = \epsilon_3/q_{\mathbf{CA}} .$$

Thus:

$$\Pr[\mathcal{B}^{\text{PRF}}(1^\lambda) = 1] = \left(1 - \Pr[\mathcal{B}^{\text{PRF}}(1^\lambda) \text{ aborts}]\right) \cdot \epsilon_3/q_{\mathbf{CA}} + \Pr[\mathcal{B}^{\text{PRF}}(1^\lambda) \text{ aborts}].$$

Clearly, $2^{-\omega}$ is negligible; moreover, if ϵ_3 is non-negligible, then so is $|\epsilon_3/q_{\mathbf{CA}}|$. Thus, to show that $|\Pr[\mathcal{B}^{\text{PRF}}(1^\lambda) = 1] - \Pr[\mathcal{B}^{\text{RAND}}(1^\lambda) = 1]|$ is non-negligible, it suffices to show that each of $\Pr[\mathcal{B}^{\text{RAND}}(1^\lambda) \text{ aborts}]$ and $\Pr[\mathcal{B}^{\text{PRF}}(1^\lambda) \text{ aborts}]$ is negligible.

To do so, recall that \mathcal{B} aborts if and only if it has previously evaluated the expression “ $\text{PRF}_{a_{\mathbf{sk},j}}^{\text{pk}}(i \| h_{\text{Sig}})$ ” using \mathcal{O} prior to receiving \mathcal{A} ’s output. First note that \mathcal{B} ’s only calls to \mathcal{O} occur when it evaluates the functions PRF^{addr} , PRF^{sn} and PRF^{pk} . Moreover, due to the construction of these functions it is not possible to evaluate the expression $\text{PRF}_{a_{\mathbf{sk},j}}^{\text{pk}}(i \| h_{\text{Sig}})$ using any calls to PRF^{addr} or PRF^{sn} . Thus \mathcal{B} aborts if and only if it has previously queried PRF^{pk} on the expression $\text{PRF}_{a_{\mathbf{sk},j}}^{\text{pk}}(i \| h_{\text{Sig}})$. However it is easy to see that this cannot happen under the conditions of $\text{EVENT}_{\text{mac}}$, since such a query would imply the condition $\text{EVENT}_{\text{sig}}$ or $\text{EVENT}_{\text{col}}$, each of which is excluded by $\text{EVENT}_{\text{mac}}$. Hence the probability of either condition occurring is 0.

Bounding the probability of Event_{key}. Define $\epsilon_4 := \Pr[\text{EVENT}_{\text{key}}]$, and let \mathcal{E} be the zk-SNARK extractor for \mathcal{A} . Assume by way of contradiction that ϵ_4 is non-negligible. We construct an algorithm \mathcal{B} that finds collisions for PRF^{sn} with non-negligible probability (contradicting the fact that PRF^{sn} is collision resistant). The algorithm \mathcal{B} works as follows.

1. Run \mathcal{A} (simulating its interaction with the challenger \mathcal{C}) to obtain tx^* .
 2. Run $\mathcal{E}(\text{pk}_{\text{POUR}}, \text{vk}_{\text{POUR}})$ to obtain a witness a for the zk-SNARK proof π_{POUR} in tx^* .
 3. If a is not a valid witness for the instance $x := (\text{rt}, \text{sn}_1^{\text{old}}, \text{sn}_2^{\text{old}}, \text{cm}_1^{\text{new}}, \text{cm}_2^{\text{new}}, v_{\text{pub}}, h_{\text{Sig}}, h_1, h_2)$, abort and output 0.
 4. Parse a as $(\text{path}_1, \text{path}_2, \mathbf{c}_1^{\text{old}}, \mathbf{c}_2^{\text{old}}, \text{addr}_{\text{sk},1}^{\text{old}}, \text{addr}_{\text{sk},2}^{\text{old}}, \mathbf{c}_1^{\text{new}}, \mathbf{c}_2^{\text{new}})$.
 5. For each $i \in \{1, 2\}$, parse $\mathbf{c}_i^{\text{old}}$ as $(\text{addr}_{\text{pk},i}^{\text{old}}, v_i^{\text{old}}, \rho_i^{\text{old}}, r_i^{\text{old}}, s_i^{\text{old}}, \text{cm}_i^{\text{old}})$.
 6. For each $i \in \{1, 2\}$, parse $\text{addr}_{\text{sk},i}^{\text{old}}$ as $(a_{\text{sk},i}^{\text{old}}, \text{sk}_{\text{enc},i}^{\text{old}})$.
(Note that, since a is a valid witness, $\text{sn}_i^{\text{old}} = \text{PRF}_{a_{\text{sk},i}^{\text{old}}}^{\text{sn}}(\rho_i^{\text{old}})$ for all $i \in \{1, 2\}$.)
 7. For each $i \in \{1, 2\}$:
 - (a) Look for a pour transaction $\text{tx} \in \mathcal{T}$ that contains sn_i^{old} .
 - (b) If one tx is found, let \overline{a}_{sk} and $\overline{\rho}$ be the seed and input used to compute sn_i^{old} in tx ; thus, $\text{sn}_i^{\text{old}} = \text{PRF}_{\overline{a}_{\text{sk}}}^{\text{sn}}(\overline{\rho})$. If $a_{\text{sk},i}^{\text{old}} \neq \overline{a}_{\text{sk}}$, output $((a_{\text{sk},i}^{\text{old}}, \rho_i^{\text{old}}), (\overline{a}_{\text{sk}}, \overline{\rho}))$ as a collision for PRF^{sn} .
- Note that, whenever $\text{EVENT}_{\text{key}}$ holds:
- the proof π_{POUR} is valid and, with all but negligible probability, the witness a is valid;
 - the serial number sn_1^{old} or sn_2^{old} appears in some previous pour transaction in \mathcal{T} ;
 - whenever a is valid, it holds that $h_1 = \text{PRF}_{a_{\text{sk},1}^{\text{old}}}^{\text{pk}}(h_{\text{Sig}})$ and $h_2 = \text{PRF}_{a_{\text{sk},2}^{\text{old}}}^{\text{pk}}(h_{\text{Sig}})$, so that it cannot be that $a_{\text{sk},1}^{\text{old}} = a_{\text{sk},2}^{\text{old}} = \overline{a}_{\text{sk}}$ (as this contradicts the conditions of the event $\text{EVENT}_{\text{key}}$).
- Overall, we conclude that \mathcal{B} finds a collision for PRF^{sn} with probability $\epsilon_4 - \text{negl}(\lambda)$.

D.3 Proof of balance

Define $\epsilon := \text{Adv}_{\Pi, \mathcal{A}}^{\text{BAL}}(\lambda)$; our goal is to show that ϵ is negligible in λ . Recall that ADDR is the set of addresses returned by \mathcal{A} 's **CreateAddress** queries.

Augmenting the ledger with witnesses. We modify the **BAL** experiment in a way that does not affect \mathcal{A} 's view: the challenger \mathcal{C} computes, for each pour transaction tx_{Pour} on the ledger L (maintained by the oracle \mathcal{O}^{DAP}), a witness $a = (\text{path}_1, \text{path}_2, \mathbf{c}_1^{\text{old}}, \mathbf{c}_2^{\text{old}}, \text{addr}_{\text{sk},1}^{\text{old}}, \text{addr}_{\text{sk},2}^{\text{old}}, \mathbf{c}_1^{\text{new}}, \mathbf{c}_2^{\text{new}})$ for the zk-SNARK instance $x = (\text{rt}, \text{sn}_1^{\text{old}}, \text{sn}_2^{\text{old}}, \text{cm}_1^{\text{new}}, \text{cm}_2^{\text{new}}, v_{\text{pub}}, h_{\text{Sig}}, h_1, h_2)$ corresponding to tx_{Pour} .²⁹ In this way, \mathcal{C} obtains an *augmented ledger* (L, \vec{a}) , where a_i is a witness for the zk-SNARK instance x_i of the i -th pour transaction in L . Note that we can parse (L, \vec{a}) as a list of matched pairs $(\text{tx}_{\text{Pour}}, a)$ where tx_{Pour} is a pour transaction in L and a is its corresponding witness.

The discussion below is relative to the above modification of the **BAL** experiment.

Balanced ledgers. We say that an augmented ledger (L, \vec{a}) is *balanced* if the following holds.

- I. Each $(\text{tx}_{\text{Pour}}, a)$ in (L, \vec{a}) contains openings (i.e., decommitments) of two distinct coin commitments cm_1^{old} and cm_2^{old} ; also, each cm_i^{old} is the output coin commitment of a pour or mint transaction that precedes tx_{Pour} on L .
- II. No two $(\text{tx}_{\text{Pour}}, a)$ and $(\text{tx}'_{\text{Pour}}, a')$ in (L, \vec{a}) contain openings of the same coin commitment.

²⁹ Concretely, for pour transactions in L not inserted by \mathcal{A} , \mathcal{C} simply retains the witness a internally used by \mathcal{O}^{DAP} to generate the transaction. As for the (valid) pour transactions inserted by \mathcal{A} , \mathcal{C} uses the zk-SNARK multi-instance knowledge extractor corresponding to \mathcal{A} ; see Section 2.1. (If knowledge extraction fails, \mathcal{C} aborts and outputs 1. However, this only happens with negligible probability.)

III. Each $(\text{tx}_{\text{Pour}}, a)$ in (L, \vec{a}) contains openings of $\text{cm}_1^{\text{old}}, \text{cm}_2^{\text{old}}, \text{cm}_1^{\text{new}}, \text{cm}_2^{\text{new}}$ to values $v_1^{\text{old}}, v_2^{\text{old}}, v_1^{\text{new}}, v_2^{\text{new}}$ (respectively), with the condition that $v_1^{\text{old}} + v_2^{\text{old}} = v_1^{\text{new}} + v_2^{\text{new}} + v_{\text{pub}}$.

IV. For each $(\text{tx}_{\text{Pour}}, a)$ in (L, \vec{a}) and for each $i \in \{1, 2\}$, the following conditions hold:

- (a) If cm_i^{old} is also the output of a mint transaction tx_{Mint} on L , then the public value v in tx_{Mint} is equal to v_i^{old} .
- (b) If cm_i^{old} is also the output of a pour transaction tx'_{Pour} on L , then its witness a' contains an opening of cm_i^{old} to a value v' that is equal to v_i^{old} .

V. For each $(\text{tx}_{\text{Pour}}, a)$ in (L, \vec{a}) , where tx_{Pour} was inserted by \mathcal{A} , it holds that, for each $i \in \{1, 2\}$, if cm_i^{old} is the output of an earlier mint or pour transaction tx' , then the public address of the i -th output of tx' is not contained in ADDR .

Intuitively, the above conditions ensure that, in L , \mathcal{A} did not spend money that was not previously minted, or paid to an address under \mathcal{A} 's control. Concretely, one can prove by induction that if (L, \vec{a}) is balanced then $v_{\text{Unspent}} + v_{\text{Basecoin}} + v_{\mathcal{A} \rightarrow \text{ADDR}} > v_{\text{Mint}} + v_{\text{ADDR} \rightarrow \mathcal{A}}$.

In light of the above, it suffices to argue that the augmented ledger induced by the (modified) BAL experiment is balanced with all but negligible probability. Suppose, by way of contradiction, that is is not the case: \mathcal{A} induces, with non-negligible probability, an augmented ledger (L, \vec{a}) that is *not* balanced. We distinguish between five cases, corresponding to which one of the above conditions does not hold with non-negligible probability. In each case, we show how to reach a contradiction, concluding the proof.

\mathcal{A} violates Condition I. Suppose that $\Pr[\mathcal{A} \text{ wins but violates Condition I}]$ is non-negligible. By construction of \mathcal{O}^{DAP} , every $(\text{tx}_{\text{Pour}}, a)$ in (L, \vec{a}) for which tx_{Pour} was not inserted by \mathcal{A} satisfies Condition I; thus, the violation can only originate from a pair $(\text{tx}_{\text{Pour}}, a)$ in (L, \vec{a}) for which tx_{Pour} was inserted by \mathcal{A} and such that: (i) $\text{cm}_1^{\text{old}} = \text{cm}_2^{\text{old}}$; or (ii) there is $i \in \{1, 2\}$ such that cm_i^{old} has no corresponding output coin commitment in any pour or mint transaction that precedes tx_{Pour} on L .

Observe that the validity of tx_{Pour} implies that:

- The two serial numbers sn_1^{old} and sn_2^{old} are distinct. Moreover, recalling that each sn_i^{old} equals $\text{PRF}_{a_{\text{sk},i}^{\text{old}}}^{\text{sn}}(\rho_i^{\text{old}})$, this also implies that $(a_{\text{sk},1}^{\text{old}}, \rho_1^{\text{old}}) \neq (a_{\text{sk},2}^{\text{old}}, \rho_2^{\text{old}})$.
- The witness a contains two valid authentication paths $\text{path}_1, \text{path}_2$ for a Merkle tree constructed using only coin commitments of transactions preceding tx_{Pour} in L .

In either (i) or (ii), we reach a contradiction. Indeed:

- (i) If $\text{cm}_1^{\text{old}} = \text{cm}_2^{\text{old}}$, then the fact that $\text{sn}_1^{\text{old}} \neq \text{sn}_2^{\text{old}}$ implies that the witness a contains two distinct openings of cm_1^{old} (the first opening contains $(a_{\text{sk},1}^{\text{old}}, \rho_1^{\text{old}})$, while the second opening contains $(a_{\text{sk},2}^{\text{old}}, \rho_2^{\text{old}})$). This violates the binding property of the commitment scheme COMM.
- (ii) If there is $i \in \{1, 2\}$ such that cm_i^{old} does not previously appear in L , then path_i is an invalid authentication path, and thus yields a collision in the function CRH. This violates the collision resistance of CRH.

\mathcal{A} violates Condition II. Suppose that $\Pr[\mathcal{A} \text{ wins but violates Condition II}]$ is non-negligible. Observe that, when Condition II is violated, L contains two pour transactions $\text{tx}_{\text{Pour}}, \text{tx}'_{\text{Pour}}$ spending the same coin commitment cm , and revealing two serial numbers sn and sn' . Since $\text{tx}_{\text{Pour}}, \text{tx}'_{\text{Pour}}$ are valid, it must be the case that $\text{sn} \neq \text{sn}'$. However (as argued already above), if both transactions spend cm but produce different serial numbers, then the corresponding witnesses a, a' contain different openings of cm . This contradicts the binding property of the commitment scheme COMM.

\mathcal{A} violates Condition III. Suppose that $\Pr[\mathcal{A} \text{ wins but violates Condition III}]$ is non-negligible. In this case, the contradiction is immediate: whenever Condition III is violated, the equation

$v_1^{\text{old}} + v_2^{\text{old}} = v_1^{\text{new}} + v_2^{\text{new}} + v_{\text{pub}}$ does not hold, and thus, by construction of the statement POUR , the soundness of the zk-SNARK is violated as well.

\mathcal{A} violates Condition IV. Suppose that $\Pr[\mathcal{A} \text{ wins but violates Condition IV}]$ is non-negligible. Observe that, when Condition IV is violated, L contains:

- a pour transaction tx_{Pour} in which a coin commitment cm^{old} is opened to a value v^{old} ; and also
- a (mint or pour) transaction tx' that opens cm^{old} to a value v' different from v^{old} .

This contradicts the binding property of the commitment scheme COMM .

\mathcal{A} violates Condition V. Suppose that $\Pr[\mathcal{A} \text{ wins but violates Condition V}]$ is non-negligible. Observe that, when Condition V is violated, L contains an inserted pour transaction tx_{Pour} that spends the output of a previous transaction tx' whose public address $\text{addr}_{\text{pk}} = (a_{\text{pk}}, \text{pk}_{\text{enc}})$ lies in ADDR ; moreover, the witness associated to tx' contains a_{sk} such that $a_{\text{pk}} = \text{PRF}_{a_{\text{sk}}}^{\text{addr}}(0)$. We omit the full argument, but one can verify that, in this case, we can construct a new adversary \mathcal{B} that uses \mathcal{A} to distinguish, with non-negligible probability, PRF from a random function.

References

- [BB04] Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In *Proceedings of the 24th Annual International Cryptology Conference*, CRYPTO '04, pages 443–459, 2004.
- [BBDP01] Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. Key-privacy in public-key encryption. In *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security*, ASIACRYPT '01, pages 566–582, 2001.
- [BBSU12] Simon Barber, Xavier Boyen, Elaine Shi, and Ersin Uzun. Bitter to better - how to make Bitcoin a better currency. In *Proceedings of the 16th International Conference on Financial Cryptography and Data Security*, FC '12, pages 399–414, 2012.
- [BCCT12] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 326–349, 2012.
- [BCCT13] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. Recursive composition and bootstrapping for SNARKs and proof-carrying data. In *Proceedings of the 45th ACM Symposium on the Theory of Computing*, STOC '13, pages 111–120, 2013.
- [BCG⁺13] Eli Ben-Sasson, Alessandro Chiesa, Daniel Genkin, Eran Tromer, and Madars Virza. SNARKs for C: verifying program executions succinctly and in zero knowledge. In *Proceedings of the 33rd Annual International Cryptology Conference*, CRYPTO '13, pages 90–108, 2013.
- [BCGT13a] Eli Ben-Sasson, Alessandro Chiesa, Daniel Genkin, and Eran Tromer. Fast reductions from RAMs to delegatable succinct constraint satisfaction problems. In *Proceedings of the 4th Innovations in Theoretical Computer Science Conference*, ITCS '13, pages 401–414, 2013.
- [BCGT13b] Eli Ben-Sasson, Alessandro Chiesa, Daniel Genkin, and Eran Tromer. On the concrete efficiency of probabilistically-checkable proofs. In *Proceedings of the 45th ACM Symposium on the Theory of Computing*, STOC '13, pages 585–594, 2013.
- [BCI⁺13] Nir Bitansky, Alessandro Chiesa, Yuval Ishai, Rafail Ostrovsky, and Omer Paneth. Succinct non-interactive arguments via linear interactive proofs. In *Proceedings of the 10th Theory of Cryptography Conference*, TCC '13, pages 315–333, 2013.
- [BCTV14] Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, and Madars Virza. Succinct non-interactive zero knowledge for a von Neumann architecture. In *Proceedings of the 23rd USENIX Security Symposium*, Security '14, pages ???–???, 2014. Available at <http://eprint.iacr.org/2013/879>.
- [Bel06] Mihir Bellare. New proofs for NMAC and HMAC: security without collision-resistance. In *Proceedings of the 26th Annual International Conference on Advances in Cryptology*, CRYPTO '06, pages 602–619, 2006.
- [Ben13] Eli Ben-Sasson. Universal and affordable computational integrity, May 2013. Bitcoin 2013: The Future of Payments. URL: <http://www.youtube.com/watch?v=YRcPReUpkCU&feature=youtu.be&t=26m6s>.
- [BFLS91] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, STOC '91, pages 21–32, 1991.
- [BGH⁺05] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil Vadhan. Short PCPs verifiable in polylogarithmic time. In *Proceedings of the 20th Annual IEEE Conference on Computational Complexity*, CCC '05, pages 120–134, 2005.
- [Cer00] Certicom Research. SEC 1: Elliptic curve cryptography, 2000. URL: http://www.secg.org/collateral/sec1_final.pdf.
- [Cha82] David Chaum. Blind signatures for untraceable payments. In *Proceedings of the 2nd Annual International Cryptology Conference*, CRYPTO '82, pages 199–203, 1982.
- [CHL05] Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya. Compact e-cash. In *Proceedings of the 24th Annual International Conference on Theory and Applications of Cryptographic Techniques*, EUROCRYPT '05, pages 302–321, 2005.
- [CL01] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *Proceedings of the 20th Annual International Conference on Theory and Application of Cryptographic Techniques*, EUROCRYPT '01, pages 93–118, 2001.
- [DDM03] George Danezis, Roger Dingledine, and Nick Mathewson. Mixminion: Design of a type III anonymous remailer protocol. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, SP '03, pages 2–15, 2003.

- [DFKP13] George Danezis, Cedric Fournet, Markulf Kohlweiss, and Bryan Parno. Pinocchio Coin: building Zerocoins from a succinct pairing-based proof system. In *Proceedings of the 2013 Workshop on Language Support for Privacy Enhancing Technologies*, PETShop ’13, 2013. URL: <http://www0.cs.ucl.ac.uk/staff/G.Danezis/papers/DanezisFournetKohlweissParno13.pdf>.
- [DMS04] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: the second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, Security ’04, pages 21–21, 2004.
- [DW13] Christian Decker and Roger Wattenhofer. Information propagation in the Bitcoin network. In *Proceedings of the 13th IEEE International Conference on Peer-to-Peer Computing*, P2P ’13, pages 1–10, 2013.
- [ES13] Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable, 2013.
- [Gen04] Rosario Gennaro. Multi-trapdoor commitments and their applications to proofs of knowledge secure under concurrent man-in-the-middle attacks. In *Proceedings of the 24th Annual International Cryptology Conference*, CRYPTO ’04, pages 220–236, 2004.
- [GGPR13] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct NIZKs without PCPs. In *Proceedings of the 32nd Annual International Conference on Theory and Application of Cryptographic Techniques*, EUROCRYPT ’13, pages 626–645, 2013.
- [GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989. Preliminary version appeared in STOC ’85.
- [GOS06a] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive Zaps and new techniques for NIZK. In *Proceedings of the 26th Annual International Conference on Advances in Cryptology*, CRYPTO ’06, pages 97–111, 2006.
- [GOS06b] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for NP. In *Proceedings of the 25th Annual International Conference on Advances in Cryptology*, EUROCRYPT ’06, pages 339–358, 2006.
- [Gro10] Jens Groth. Short pairing-based non-interactive zero-knowledge arguments. In *Proceedings of the 16th International Conference on the Theory and Application of Cryptology and Information Security*, ASIACRYPT ’10, pages 321–340, 2010.
- [GW11] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*, STOC ’11, pages 99–108, 2011.
- [KL07] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. Chapman & Hall/CRC, 2007.
- [Lee13] Timothy B. Lee. Bitcoin needs to scale by a factor of 1000 to compete with Visa. here’s how to do it. The Washington Post (<http://www.washingtonpost.com>), November 2013.
- [Lip12] Helger Lipmaa. Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments. In *Proceedings of the 9th Theory of Cryptography Conference on Theory of Cryptography*, TCC ’12, pages 169–189, 2012.
- [Lip13] Helger Lipmaa. Succinct non-interactive zero knowledge arguments from span programs and linear error-correcting codes. In *Proceedings of the 19th International Conference on the Theory and Application of Cryptology and Information Security*, ASIACRYPT ’13, pages 41–60, 2013.
- [Max13] Greg Maxwell. CoinJoin: Bitcoin privacy for the real world, August 2013. Bitcoin Forum. URL: <https://bitcointalk.org/index.php?topic=279249.0>.
- [MGGR13] Ian Miers, Christina Garman, Matthew Green, and Aviel D. Rubin. Zerocoins: Anonymous distributed e-cash from bitcoin. In *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, SP ’13, pages 397–411, 2013.
- [Mic00] Silvio Micali. Computationally sound proofs. *SIAM Journal on Computing*, 30(4):1253–1298, 2000. Preliminary version appeared in FOCS ’94.
- [MPJ⁺13] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker, and Stefan Savage. A fistful of Bitcoins: Characterizing payments among men with no names. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC ’13, pages 127–140, 2013.
- [Nak09] Satoshi Nakamoto. Bitcoin: a peer-to-peer electronic cash system, 2009. URL: <http://www.bitcoin.org/bitcoin.pdf>.
- [Nat12] National Institute of Standards and Technology. FIPS PUB 180-4: Secure Hash Standard. <http://csrc.nist.gov/publications/PubsFIPS.html>, 2012.

- [PGHR13] Bryan Parno, Craig Gentry, Jon Howell, and Mariana Raykova. Pinocchio: nearly practical verifiable computation. In *Proceedings of the 34th IEEE Symposium on Security and Privacy*, Oakland '13, pages 238–252, 2013.
- [Pol13] PolarSSL. PolarSSL. <http://polarssl.org>, Oct 2013.
- [RM11] Fergal Reid and Harrigan Martin. An analysis of anonymity in the Bitcoin system. In *Proceedings of the 3rd IEEE International Conference on Privacy, Security, Risk and Trust and on Social Computing*, SocialCom/PASSAT '11, pages 1318–1326, 2011.
- [RS12] Dorit Ron and Adi Shamir. Quantitative analysis of the full Bitcoin transaction graph. Cryptology ePrint Archive, Report 2012/584, 2012.
- [ST99] Tomas Sander and Amnon Ta-Shma. Auditable, anonymous electronic cash. In *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '99, pages 555–572, 1999.
- [Val08] Paul Valiant. Incrementally verifiable computation or proofs of knowledge imply time/space efficiency. In *Proceedings of the 5th Theory of Cryptography Conference*, TCC '08, pages 1–18, 2008.
- [Wui14] Pieter Wuille. Proposed BIP for dealing with malleability. Available at <https://gist.github.com/sipa/8907691>, 2014.