Bezop : Decentralized Ecommerce Network with Verifiable Proof Of Delivery.

A Technical Analysis On 5 Key Components Of A Secure Decentralized E-commerce Network.

By Bezop Blockchain Ltd

September 26, 2017

ABSTRACT

The world wide web is experiencing a change, not functional; of design and structure. The inevitable end of centralized proprietary services will be effectively replaced with decentralized and democratized options; elimination of unreliable and human-based trust with immutable and mathematical trust verifiable by computation; a permanent overhaul of location based address with resilient content addresses; inefficient and inflexible primitive(centralized) services will witness less demand as peer-based algorithmic technologies continue to thrive. Bitcoin, Ethereum, and other blockchain networks have proven the utility of decentralized transaction ledgers.

These public ledgers process sophisticated smart contract applications and transact crypto-assets worth tens of billions of dollars. These systems are the first instances of public accessible Open Services, where participants form a decentralized network providing useful services for credit, with no central management or trusted parties.

Decentralized applications are forced to tackle quickly challenges of "legality of purpose," handling fraud and money laundering else strict regulations will emerge if it is believed to "aid or abet breaking of the laws of the land." At the moment, blockchain technology may be widely perceived of as an easy medium for tax evasion because transactions occur secretly and cryptographically, which is the core and fundamental feature. There is no easy way for a business to adopt blockchain and prove they are tax free; hence, not ideal for some businesses. While there are many more critical issues such as scaling, security and stability, the aforementioned pose a direct threat to blockchain's survival.

Bezop is a decentralized peer to peer ecommerce order management and processing system powered by smart contracts, an autonomous buyer-seller protection service and an simple vat collection system all built on a decentralized blockchain network. Bezop aims to provide an open source and complete solution for running a successful ecommerce business online . Merchants can participate by selling products and services on their self-hosted ecommerce stores (the Bezop DOM) and broadcast orders to the network.

The Bezop market runs on a blockchain with a native erc-20 protocol token (also called "Bezop"), which miners earn on the ethereum blockchain by proof of work. Clients spend Bezop in purchasing goods and services on Bezop DOM and other integrated sites offering services and accepting payment in Bezops. Conversely one can earn Bezops by selling goods and services and or simply receiving Bezop via wallet to wallet transfer. In the future, a Bezop network will emerge where miners compete to mine blocks with sizable rewards; however, Bezop mining power will be proportional to active orders on the network. Hence, the usefulness of Bezop's mining will not be limited to maintaining blockchain consensus. This creates a powerful incentive for miners to fulfil as many orders as they can. The protocol weaves these amassed resources into a self-healing blockchain network that anybody in the world can rely on. The network achieves robustness by implementing very powerful buyer protection techniques through smart contracts. The protocol's order management portal provides adequate security, as orders are encrypted end-to-end at the client, while miners do not have access to decryption keys. Bezop works as an incentive layer on top of blockchain, which computes a proof of order for any transaction. It is especially useful for decentralizing payments, building and running distributed e-commerce portals, and implementing smart contracts.

This work:

(a) Introduces the Bezop Network, gives an overview of the protocol, and walks through several components in detail

(b) Decentralized order management and its properties

(c) Implementation on Ethereum Network & Erc-20 tokens

(d) Introduces the "proof-of-order", which allows proving a transaction with rich details of the purchase as well as verifiable proof of delivery (which is a sufficient proof) the product/service was delivered, vat collection and proof of correctness for chain of transactions

(e) Discusses use cases, connections to other systems, and how to use the protocol. Bezop is a work in progress. Active research is under way, and new versions of this paper will be announced at https://bezop.io/changes/changelog.pdf For comments and suggestions, email research@Bezop.io.

Contents

1. Introduction

Bezop is a protocol token whose blockchain runs on Proof-of-Work, where blocks

are created by miners that are confirming orders (transactions). A **proof of work** is a piece of data which is difficult (costly, time-consuming) to produce but easy for others to verify and which satisfies certain requirements. Producing a proof of work can be a random process with low probability so that a lot of trial and error is required *on average* before a valid proof of work is generated. Bezop uses the Hashcash proof-of-work system.

One application of this idea is using Hashcash as a method to preventing email spam, requiring a proof-of-work on the email's contents (including the To address), on every email. Legitimate emails will be able to do the work to generate the proof easily (not much work is required for a single email), but mass spam emailers will have difficulty generating the required proofs (which would require huge computational resources).

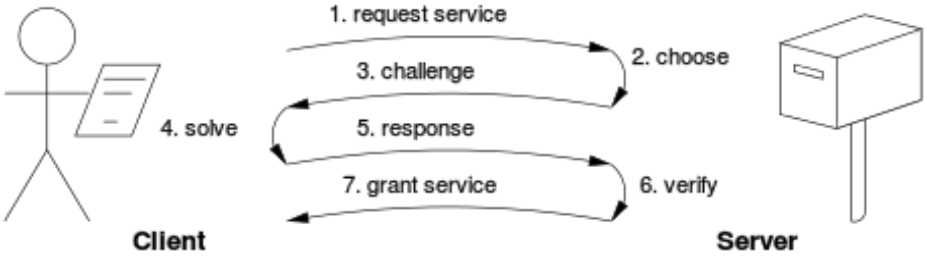
Hashcash proofs of work are used in Bezop for block generation. In order for a block to be accepted by network participants, miners must complete a proof-of-work which covers all of the data in the block. The difficulty of this work is adjusted so as to limit the rate at which new blocks can be generated by the network to one every 10 minutes. Due to the very low probability of successful generation, this makes it unpredictable which worker computer in the network will be able to generate the next block.

For a block to be valid it must hash to a value less than the current target; this means that each block indicates that work has been done generating it. Each block contains the hash of the preceding block, thus each block has a chain of blocks that together contain a large amount of work. Changing a block (which can only be done by making a new block containing the same predecessor) requires regenerating all successors and redoing the work they contain. This protects the block chain from tampering.

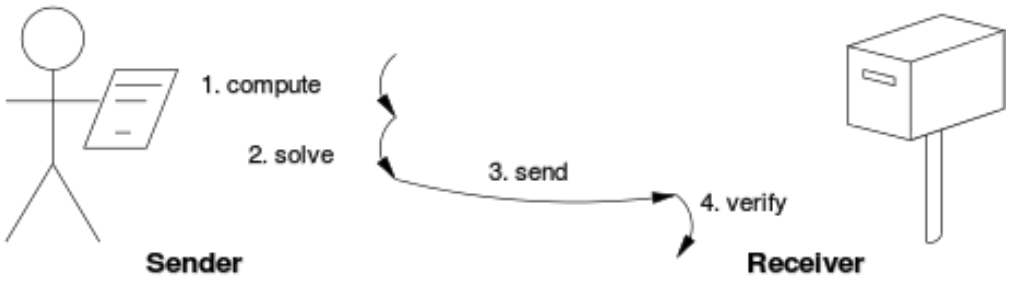
The most widely used proof-of-work scheme is based on SHA-256 and was introduced as a part of Bitcoin. Some other hashing algorithms that are used for proof-of-work include Scrypt, Blake-256, CryptoNight, HEFTY1, Quark, SHA-3, scrypt-jane, scrypt-n, and combinations thereof.

There are two classes of proof-of-work protocols.

- **Challenge-response** protocols assume a direct interactive link between the requester (client) and the provider (server). The provider chooses a challenge, say an item in a set with a property, the requester finds the relevant response in the set, which is sent back and checked by the provider. As the challenge is chosen on the spot by the provider, its difficulty can be adapted to its current load. The work on the requester side may be bounded if the challenge-response protocol has a known solution (chosen by the provider), or is known to exist within a bounded search space.



- **Solution-verification** protocols do not assume such a link: as a result the problem must be self-imposed before a solution is sought by the requester, and the provider must check both the problem choice and the found solution. Most such schemes are unbounded probabilistic iterative procedures such as Hashcash.



Known-solution protocols tend to have slightly lower variance than unbounded probabilistic protocols, because the variance of a rectangular distribution is lower than the variance of a Poisson distribution (with the same mean). A generic technique for reducing variance is to use multiple independent sub-challenges, as the average of multiple samples will have lower variance.

1.1 Elementary Components

The Bezop protocol builds upon four novel components.

i. Decentralized Order Management (DOM) : Open Source order management and fulfilment portal

ii. Bezop Network : A smart contract friendly blockchain network

iii. Proof of Order :A brief overview on the various proof's,

We present these novel Proofs-of-Order in detail (Section 3) .

a. Proof-of-Tax : - a vat transaction holds data

(i) pointing to the main transaction .

- x must be computed for correctness by taking the hash of **γ.Φ**

Tax Data : { α|obj : val , γ|str: val, δ|float : val, ε:bool:val, x|str:val}

α - data , γ - timestamp ,γ_1 - timestamp of parent transaction

δ - volume ,δ_1 - volume of parent transaction

ε - bound (inbound :1 , outbound : 0)

Φ - val($\gamma_1.\delta_1$)

i - tid of original transaction

x - correctness byte **[hash(γ.Φ)]**

b. Proof-of-Transaction : Serves a receipt, and shows all information about the product or service purchased.

Transactional Data : { α|obj : val , β|str : val, γ_1|str: val, δ_1|float : val, ε:bool:val, tid|str : val }

α - data (see page XX)

β - sender's address , γ_1 - timestamp , δ_1 - volume (amount)

ε - bound (inbound :1 , outbound : 0) , tid - transaction id

iv) Proof-of-Delivery : Proofs that a product or service was delivered.

v) Bezop Smart Contracts & Promises : Smart contracts will enable a completely safe and trustless ecommerce solution working flawlessly together on the proof of order protocol and ensuring buyer protection is made available via the blockchain.

Others :

i Correctness : correctness of an algorithm is asserted when it is said that the algorithm is correct with respect to a specification. *Functional* correctness refers to the input-output behaviour of the algorithm (i.e., for each input it produces the expected output).

A distinction is made between **total correctness**, which additionally requires that the algorithm terminates, and **partial correctness**, which simply requires that *if* an answer is returned it will be correct. Since there is no general solution to the halting problem, a total correctness

assertion may lie much deeper. A termination proof is a type of mathematical proof that plays a critical role in formal verification because total correctness of an algorithm depends on termination.

For example, successively searching through integers 1, 2, 3, … to see if we can find an example of some phenomenon—say an odd perfect number—it is quite easy to write a partially correct program (using long division by two to check *n* as perfect or not). But to say this program is totally correct would be to assert something currently not known in number theory.

Proof-of-Work :

Bezop will be based on the same POW system as Ethereum , which will further be upgraded to our full ‘proof of order’ where miners do not need to spend computation to mine blocks, but instead confirm orders on the network.

1.2 Ethereum Network , ERC20 Token :

Ethereum is a decentralized platform that runs Bezop , using the power of smart contracts provided by the Ethereum network sales of goods and services, common transactions occurs without any possibility of downtime, censorship, fraud or third party interference.

Bezop runs on a custom built blockchain (Ethereum), which is enormously powerful, transparent and a global infrastructure that can move value around and represent the ownership of property. Which will enable us to create the Bezop markets, store registries of transactions,deliveries, order status, tax and also move funds in accordance with instructions given long in the past (like a will or a futures contract) and many other things that have not been invented yet, all without a middleman or counterparty risk.

1.3 Protocol Overview :

• The Bezop protocol is a decentralized ecommerce network built on a blockchain and with a native token. Clients spend tokens in exchange for goods, services, triggering contracts and retrieving tax proof.

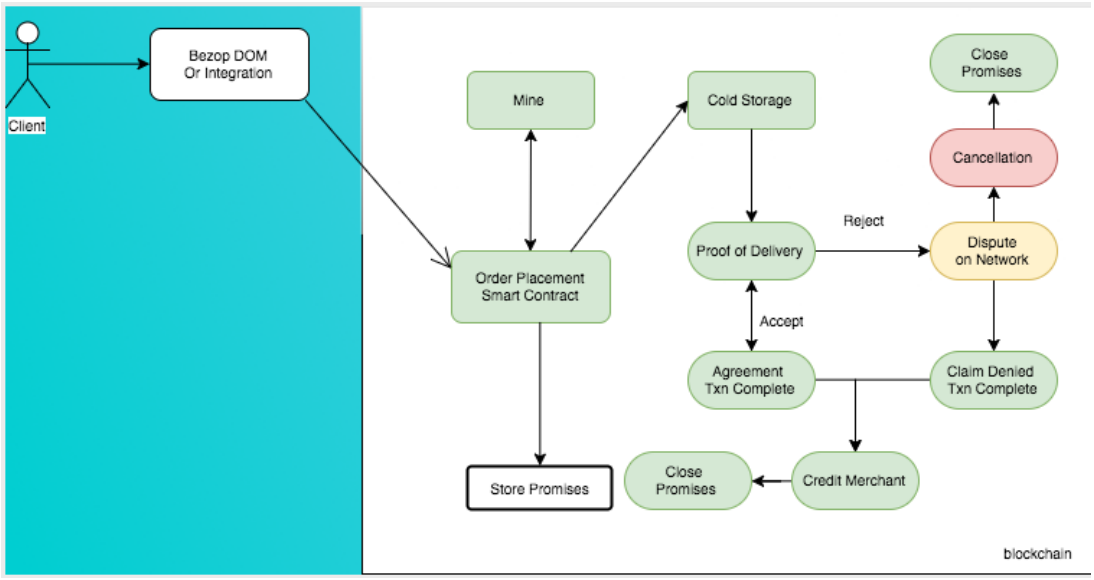
• Miners earn by confirming orders (transactions) and resolving disputes.

• The Bezop DOM handle orders on the frontend. Store Owners set the price for their product/service and VAT rate and add them to their self operated Bezop platform, Customers send tokens which serve as credit for the goods and service.

• The markets are operated by the Bezop network which employs Proof-of-Work and Proof-of-Order to guarantee that miners have correctly done work to confirm the order.

• Finally, miners can participate in the creations of new blocks for the underlying blockchain. The influence of a miner over the next block is proportional to the amount of active orders in the network.

Protocol Overview



1.3 Paper organization

The remainder of this paper is organized as follows. We present our definition of and requirements for a DOM in **Section 2**.

In **Section 3,** we motivate, define, and present our Proof-of-Order, (Proof Of Transaction and Proof-of-Tax), used within Bezop

In **Section 4,** we provide a brief description of Smart Contracts within the Bezop.

We conclude with a discussion of future work in **Section 5**.

2.1 Definition of a Decentralized Order Management

Decentralized Order Management (DOM) is a fully developed ecommerce platform which any interested user can download , host, and run without any central governance or external control.

Users can run a pull request on the Bezop's DOM from github to their self-operated web server . They can start selling their products and services in exchange for Bezop tokens in a matter of minutes.

2.2 Technologies

Bezop DOM is built with HTML5, ReactJS , NodeJs , MongoDB

Which are all open source languages.

2.2 Requirements

Running Bezop DOM requires installation of the following dependencies

- Linux
- Nginx
- NodeJs
- Npm (Node Package Manager)
- MongoDB

2.3 Properties (Described by Real-Life Case Analysis)

(i) Store Creation :

Problem Analysis A - Client 0:

Jane is the owner of a small business where she sells Sports outfit and kits for USD 10 - 150. She wanted to expand and build her own digital store where she can easily transact with her clients online without the frustration of losing massive fees per transaction or monthly fees to access a centralized portal.

Bezop Solutions A :

With Bezop DOM , Jane can instantly run a fully decentralized e-commerce portal. Without the need for PrestaShop , Shopify , Amazon and other centralized e-commerce stores that requires heavy transaction and monthly payments.

Problem Analysis B - Client 1:

John is the owner of a medium scale business and sells electronics for USD 99 - 1999. he recently got an online store but in constant dispute with his merchant service providers. He waits 21 days for credits to clear into his account and prefers not to wait that long to send funds back to his suppliers.

Bezop Solutions B :

With Bezop DOM , John is not governed by any central body and can process their own transaction by linking the Bezop main address and Bezop tax address into Bezop DOM online store. His funds are automatically clears few days after proof of delivery.

Product Addition (Limitless by Design)

Problem Analysis A - Client 2:

Jude is the owner of a Large scale business and Owns a supermarket where the sell hundreds of items of different prices USD 99 - 1999. He has an online store but is limited to only display 200 of his products at a time.

Bezop Solution A :

Bezop DOM is optimized to allow as much products as Jude desires with low memory footprint and storage requirement.

(ii) Product Categorization

Problem Analysis A - Client 3:

Ray is the owner of a Large scale business and where products sold vary by category. He needs a simple solution to start selling online while implementing this structure without handling the inherent complexity in design or programming.

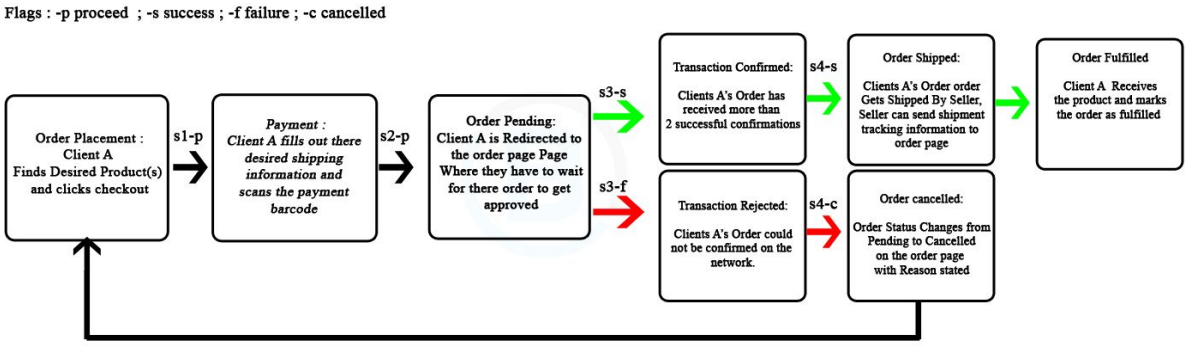
Bezop Solution A :

Bezop DOM is optimized to allow Raymond to categorize and segment his store into multiple departments which users can access through an auto-generated menu. With centralized system, you will need a content management plugin, writing codes, and configuring databases in order to achieve the same effect.

(iii)Order Management

Bezop Decentralized Order Management Portal (DOM) provides an ideal order management system with status codes indicating the order's current stage and flags representing state changes.

See Fig 2 (Figure Showing Order State)



(s1-p) : stage1 -proceed

This status message indicates a 1-step procession from order placement , in this stage the buyer is now ready to provide their email address and shipping information for delivery of their product. Likewise checkout

(s2-p) : stage2 -proceed

This status message indicates a 2-step processions from order placement, in this stage the buyer has successfully paid for the product/service and are now redirected to the order page where they must for their orders to be confirmed on the network.

(s3-s) : stage3 -success

This status message indicates a 2-step processions from order placement and 1-success transition, in this stage the buyer's transaction has received more than 2 successful confirmations.

(s3-f) : stage4 -failure

This status message indicates a 2-step processions from order placement and 1-failure transition, in this stage the buyer's transaction has not received any successful confirmations and failed .

(s4-s) : stage4 -success

This status message indicates a 2-step processions from order placement and 2-success transitions, in this stage the buyer's order has been shipped by the store owner.

(s4-c) : stage4 -cancelled

This status message indicates a 2-step procession from order placement and 1-failure transition and order cancellation, in this stage the buyer's transaction has now been cancelled because their transaction failed.

The buyer will be provided a link back to their cart. In this scenario, they need to reorder.

Order Fulfilment

This is the last stage and is a positive agreement between the buyer and seller that the transaction is complete.

(iv) Support Messages

Bezop DOM will allow customers to exchange messages with seller after their order has been placed ,

This will allow both parties to stay in sync.

(v) Generation Of Proof Of Tax Chain

A Pioneer feature amongst many is the ability to collect VAT with BEZOP and segment funds into two addresses; a secondary address (Tax address) and a primary (Main Address) - both addresses are entangled. The proof of tax chain makes it possible to generate a comparison table of all valid transaction splits between the Tax Address and the Main Address.

Trigger_contract("PoTxM","PoTxV") //Function returns Xm,Xv

XM ; XV ; I:INT=0 ; XM:ARRAY=[T1,T2,T3,..,TN] //holds transaction history

XV:ARRAY=[T1,T2,T3,..,TN] //vat history

WHILE I<COUNT(XM){PRINT XM[I]}

WHILE I<COUNT(XV){PRINT XM[I] ; PRINT <hr />}

Bezop Wallet Integration

Based on smart contract, Bezop Dom will allow a merchant Bezop Wallet seamless integration, easy receipt and ultra fast transaction processing. With the help of the Ethereum blockchain, we are able to clear transactions into a Bezop wallet with 1 step and 1 fee, which results in accepting payments generally up to 5X cheaper and faster for merchants. Merchants will be able to accept payments from Ethereum-based wallets.

3) Proof Of Order

Allows proving a transaction occurred. In the case of a purchase, it must provide (sufficient proof) the service was delivered, a valid proof of tax (vat collection) and proof of correctness for a single or chain of transactions.

The proof of order adds ‘legitimacy’ and ‘legality of purpose’ to every transaction on the Bezop network while remaining decentralized, anonymous and trustless.

Consensus for complete ‘Proof of Order’

- Sellers must convince the network that they have delivered the paid products, service or software in good condition; Sellers will generate Proofs-of-Delivery that the blockchain network may verify
- Blockchain Network must store proof of transaction
- Blockchain Network may generate proof of tax

In this section, we motivate, present and outline implementations for the Proof-of-Delivery (PoD) and Proof-of-Transaction (PoT) and Proof-of-Tax (PoT) schemes used in Bezop.

Motivation

Proofs-of-Order schemes such as Proof-of-Delivery (POD) and Proof-of-Transaction (PoT) schemes allow the network (N) to track present state and store transaction data a smart contract protocol with the user.

POD scheme guarantees that an order is delivered.

Attacks

In Bezop, we need stronger guarantees to prevent three types of attacks that malicious

sellers could exploit to get paid for products and services they are not providing:

Rogovy attack,

Permanent Eclipse Attack

Rogovy Attack:

This is a kind of attack where several merchants turn rogue and mark large number of orders as delivered while the actual delivery is not as described. In this case a buyer ends up paying for what they cannot use.

Components for Bezop Rogovy Attack

The attacker(s) must own multiple merchant addresses and rapidly trigger Proof-of-Delivery Smart Contract Protocol with the aid of a bot.

The faith of this attack falls on the client's ability to respond in timely fashion.

Permanent Eclipse Attack :

This is a consensus critical vulnerability in the Ethereum peer-to-peer protocol. If the vulnerability is exploited an eclipse attack could cause Denial-of-Service and that can also be used by an adversary to double spend transactions. Due to the low resource requirements of the attack, an attacker with limited capabilities can easily sustain an attack on the whole ethereum network (ultimately Bezop).

Components for Bezop-Ethereum Eclipse Attack

The attack appears to be a vulnerability in the peer-to-peer protocol. However, it was only tested with the geth client.

- **Block Propagation**

Blocks are propagated in three different ways in the Ethereum network. Firstly, a node B that receives a new block, will directly push the block to $\sqrt{n}$ of its connected peers, where n is the total number of connected peers. Secondly, B will send a NewBlockHashes message to all of its peers, advertising a new block.

When a node A receives an advertisement, it will request the block explicitly after 0.5 seconds from a random peer from which it received a corresponding advertisement (unless A received the block from another peer in the meantime) and then forgets about all other advertisements for that block. This means that if A requests the block from B and B fails to answer, it will not request the block from any other peers. If A misses a block, the third method for block propagation is used, which is the block synchronisation explained below in section 1.3.2.

- **Block Synchronisation**

A node will only synchronise with one other node at a time. A node "A" starts a block synchronisation in the following cases:

"A" starts a connection to a new peer with higher advertised total difficulty (e.g. after joining or rejoining the network).

"A" node advertising a higher total difficulty than "A" connects to "A".

"A" receives a block with higher total difficulty than the head of its current blockchain and is missing some of the blocks ancestors

- **Attack Details**

The following is a consensus critical flaw in the peer-to-peer protocol. Using this vulnerability, an attacker B can keep a victim A from receiving a block at height n + 1 almost indefinitely. While A may receive later blocks from other peers, it will be stalled at height 'n' since it misses a link in the blockchain. An attacker B can work as follows:

1. B creates a long blockchain starting from the genesis block. This can be done by decreasing the difficulty for each block, thus shortening the time needed to generate a valid proof-of-work.

Creating the alternative blockchain takes a relatively large amount of precomputation. However, once the alternative blockchain exists, it does not need to be computed again and can be used for multiple attacks.

2. If B's chain is shorter than the valid chain, B forges a block with a high block number and valid proof-of-work, i.e., it creates a block without parent that uses an arbitrary value as parent hash.

3. If the attacker blockchain is shorter than the valid blockchain, B also needs to forge a block with block number MaxHeaderFetch below the height of A's blockchain and blocks at heights that will be queried in the binary search but are not contained in B's blockchain.

4. B connects to Node A, advertising a high total difficulty.

5. B sends the block header with the highest block number from his chain (or the block created in step 2) to A in response to the first GetBlockHeaders request from A.

6. In response to the second GetBlockHeaders request from A, B sends the block from its separate blockchain corresponding to the block number specified in A's message (or the block forged in step 3).

7. In response to the requests from A's binary search, B responds again with block headers from its own blockchain until the genesis block is reached.

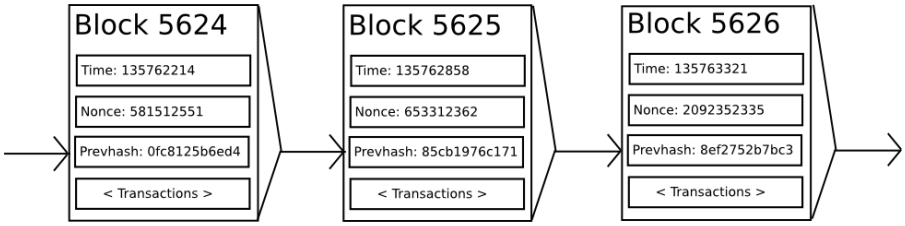
8. A will now request blocks starting from the genesis block. B responds with blocks from its own blockchain, while sending only one block in response for each request. For each request, B can introduce a delay of up to 3 seconds.

9. As long as the attack is in progress, A will not be able to synchronise with any other peers.

Proof of work :

The mechanism behind proof of work was a breakthrough 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 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 its computational resources. The discussion concerning 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.

Bezop Mining



If we had access to a trustworthy centralized service, this system would be trivial to implement; it could 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 create one block approximately every ten minutes, with each block containing a timestamp, a nonce, a reference to (i.e., 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 continually 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:

- Check if the previous block referenced by the block exists and is valid.
- Check that the timestamp of the block is greater than that of the previous block and less than 2 hours into the future
- Check that the proof of work on the block is valid
- Let S[0] be the state at the end of the previous block
- 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
- 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 2187. 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.

For instance,a target of ~2187, the network must make an average of ~269 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 270,000. 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 bitcoin blockchain, starting by mining another version of block 270,000 pointing to the same block 269,999 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 for the concerned block. Furthermore, the attacker's new version of block 270,000 has a different hash, so the original blocks 270,001 to 270,005 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 270005 chain while the attacker alone is working on the 270,000 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"). Bitcoin blocks depend on the hash of all previous blocks. An attacker with immense computing power can redo the proof of work (PoW) for a considerate amount of blocks and can eventually gain a lot of bitcoins but as described in Satoshi's paper,the reward to mine a valid block is much more than to disrupt the network. But in light of falling mining rewards the same does not hold true.

Proof Of Transaction and Delivery

PoT : A digital '**Proof of transaction**' is generated by the Bezop network for any and all sale of goods and services, it is publicly viewable, immutable and preserves anonymity for both parties.

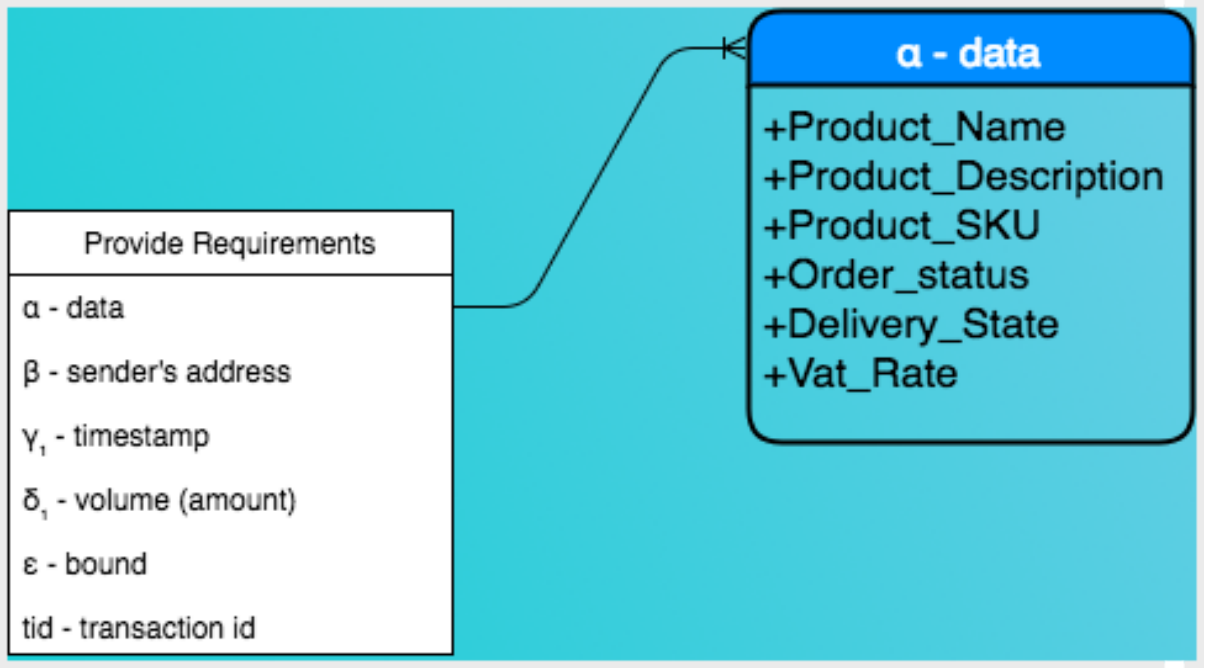
Transactional Data : { α|obj: val , β|str : val, γ_1|str: val, δ_1|float : val, ε:bool:val, tid|str : val }

α - data , β - sender's address , γ_1 - timestamp , δ_1 - volume (amount)

ε - bound (inbound :1 , outbound : 0) , tid - transaction id

α - data is an object containing
Product_Name
Product_Description
Product_SKU
Order_status
Delivery_State
Vat_Rate

Fig Proof Of Transaction (Data Structure)



PoD : Proof of delivery (POD) is a method to establish the fact that the recipient received the contents sent by the sender. When the sender sends multiple physical goods or digital content, there is a possibility of some not reaching the intended recipient.

Proof of delivery becomes very important when legal and financial documents are to be exchanged between two parties. In the United States, DHL, UPS and FedEx as well as the US postal service (USPS) provide proof of delivery. Commercial fleet operators also need to be able to confirm proof of delivery of goods to their customers.

In e-commerce, businesses exchange millions of electronic documents to track delivery information using computer to computer communication techniques like email, FTP and EDI. These documents contain a variety of transaction details, including information regarding purchase orders, invoices, shipping details, product specifications, and price quotes. Electronic documents can exchange new data as well as corrections to previously transmitted messages.

Legal complications can arise if the recipient's company refutes receiving a corrected product specification or a message about a delayed shipment. Both companies could be at loggerheads, each proving/not proving the existence of that particular communication.

Decentralized proof of delivery is similar in function however works in reverse, the shipper/store owner Is liable to provide a 'proof of delivery' within 30days of marking the order as shipped on the network, this can be done through the DOM which is linked to the receiving wallet ; the options are

a) Providing a link to scanned delivery document
b) Providing link to proof of delivery screenshot
c) tracking number of the package

Providing proof of delivery requires a gas fee which Miners will use to confirm the delivery. Once delivery is approved by a miner, the order state is changed to " has been provided", the order is deemed fulfilled, and order_status is updated.

Proof Of Tax

A digital '**Proof of tax**' is generated and stored on the Bezop network for transactions related to sales of goods and services. Bezop's DOM provides a powerful feature which allows users to collect VAT for their products to their secondary wallet address which the network automatically issues to all Bezop account holders. A proof of tax is publicly viewable, immutable and preserves anonymity for both parties.

A tax wallet operates just like a standard wallet with the exception that it is entangled/bonded to the main address using an easily computable hash known as the bond-derivative. The purposes a tax wallet serves are listed below :

1) Collecting Value Added Tax (VAT)
2) Automatic calculation and generation of tax report

A tax report stored on blockchain is immutable and resistant to loss

Cases

Your Accountant Moves or Closes

Situation: Yesterday, I found myself in quite a predicament when I couldn't get in contact with my old accountant. Turns out his office had closed business just when I needed him to give me a copy tax return to refinance my home. I had been careless and hadn't saved a copy, and had never been really bothered that I hadn't because I always thought my CPA was my backup for maintaining copies of my past tax return and my insurance for that… too bad I was wrong. So what do you do if you find yourself in a similar situation and your tax coach is missing in action?

YOU move and threw away all your papers or just shredded your last tax return copy
Situation: I recently relocated to be closer to my family and my loved ones, but realized that I threw away my last year's tax return copy. I thought for some reason I wouldn't need it, and how my tax preparer is far away and not picking up. My son is going to college and applying for financial aid-I need it to help him get settled here and so I have a piece of mind. How do I get a copy of an old tax return quickly and securely?

You lost your copy of your tax return in a fire or flood
Situation: My family and I just went through a harrowing ordeal of losing our home to a brush fire. Everyone is safe and no one was hurt, but now I have to build my life all over from scratch and I don't even know where to begin. I know I need to get my finances together and to do that I need a copy of my tax returns. I know there's some way to get a copy without having to wait so long for the IRS to send them - I'm homeless and scared I want someone to help me right now! Is there any way I can get a copy of my past tax return and the help that I need?

Your computer crashed with all your copies of tax returns download
Situation: I'm a grad student who is detailed, studious, and organized. But yesterday my computer suddenly crashed with a virus and now I have no access to my tax return copies I filed and saved electronically! I need my financial documents for the bar exam, for my own recordkeeping, and the new lease application that's due tomorrow! Help! Where can I go to ask for help to get a copy of my old tax return at 3AM in the morning?

3) Smart Contracts

3.1 Contracts in Bezop : Smart contracts will enable a completely safe and trustless e-commerce solution on the proof of order protocol and ensuring buyer protection is made available via the ethereum blockchain.

A number of smart contracts, detailed structure and functionality will be defined in this section and finalized with more research. To illustrate the intended functionality, we provide sample workflows illustrating the purchase, cancellation, claim process, and dispute handling on the blockchain.

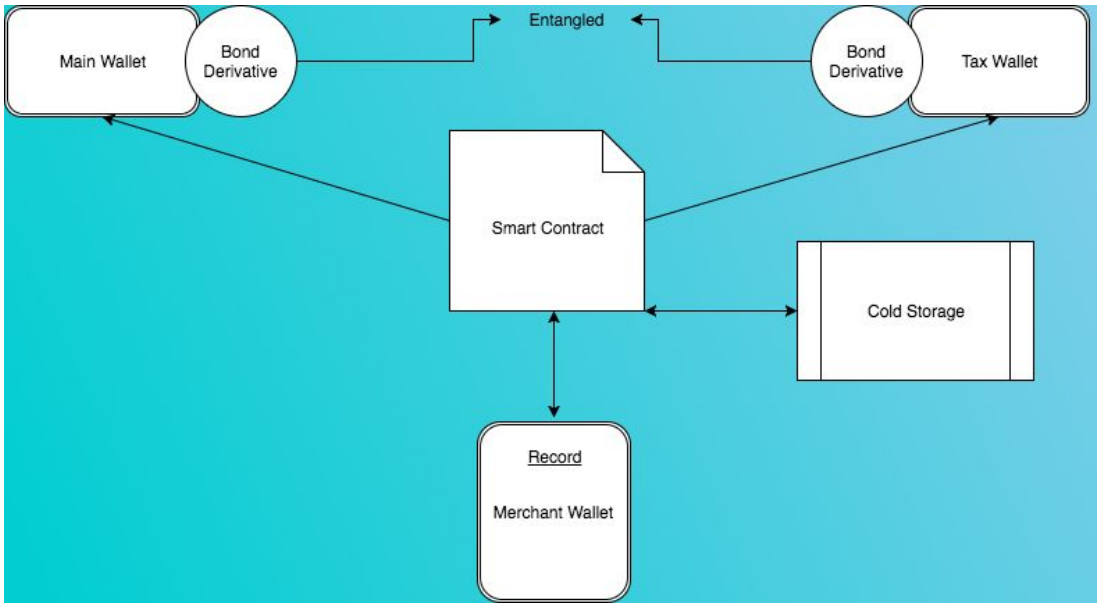
Merchant Wallet (powered by smart contract):

Buyer protection is a critical problem in e-commerce. Bezop DOM (decentralized order management) system is designed to only work with specialized wallets. A standard Bezop wallet is comprised of a

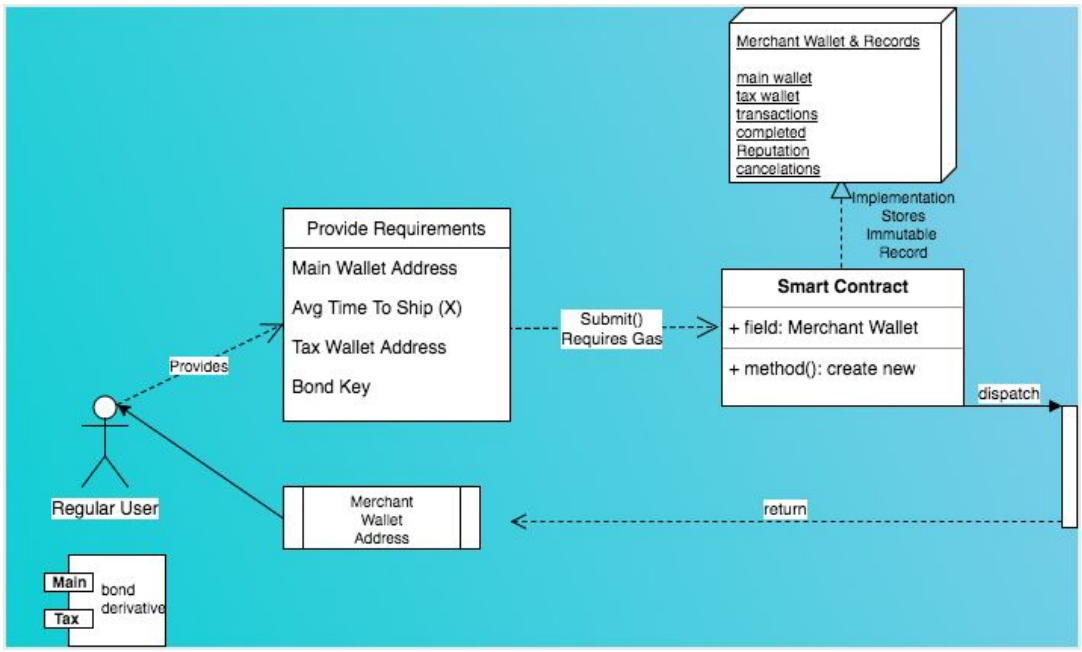
“main wallet” and a “tax wallet” entangled using an easily computable hash known as “bond-derivative”.

The Merchant wallet is a simple smart contract on the blockchain that does the work of a middle-man (escrow system) and keeps both the client and merchant safe. The contract transfers a client’s funds into cold storage. Which is unlockable only after (X) days when proof-of-delivery is provided by the merchant. If a merchant fails to provide PoD, the buyer will have the option to cancel.

Simple Figure shows the merchant/contract wallet design, how the main wallet and tax wallet are linked with a bond-derivative)



Merchant Wallet Creation (smart contract)



A merchant wallet simulates the behaviour of a normal wallet however it is based on a contract address and operated autonomously.

X signifies the number of days before shipment can be made. This is specified by the merchant while creating their merchant wallet.

Order Placement (Smart Contract)

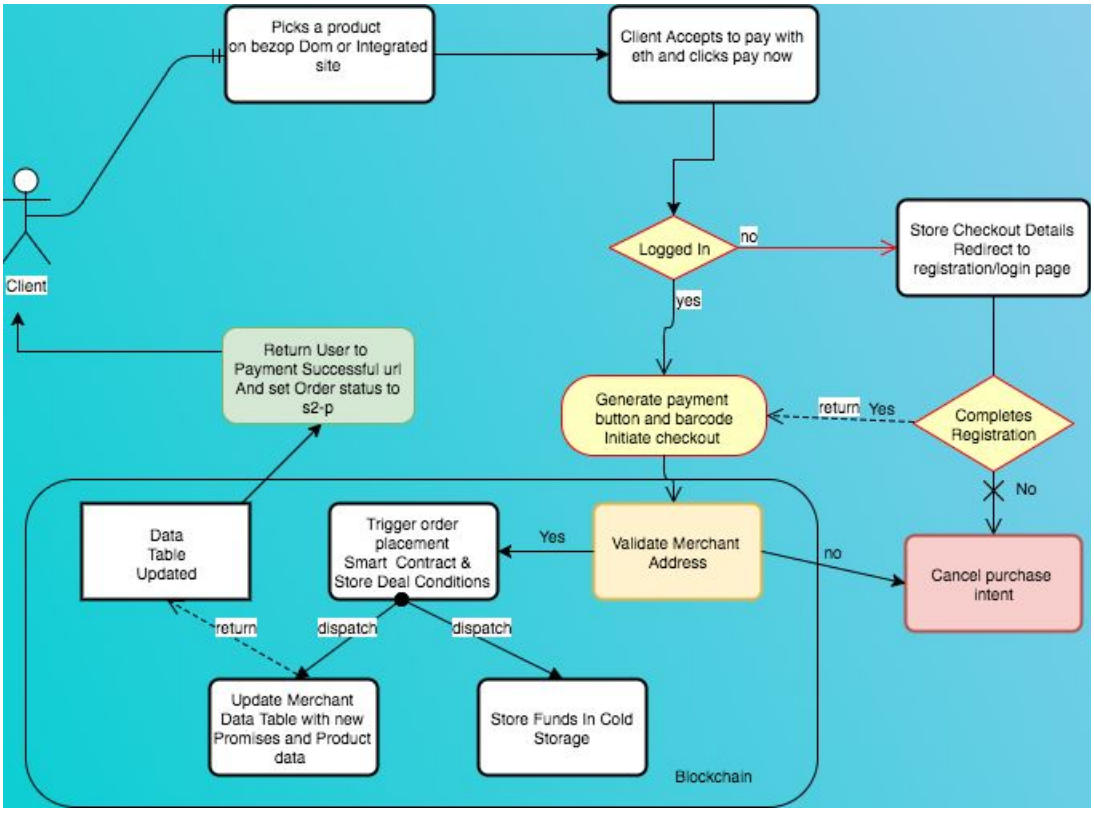
During order placement, tokens are sent to a merchant's wallet address (contract on the Ethereum network), this triggers a smart contract which automatically transfers the funds to a cold secure storage wallet and in place , stores a digital IOU to the registry known as "a Promise record". This promise to credit the main wallet and tax wallet only holds after all conditions have been met for the sale. Some generic conditions are :

i) X days has passed

ii) Proof of Delivery is provided

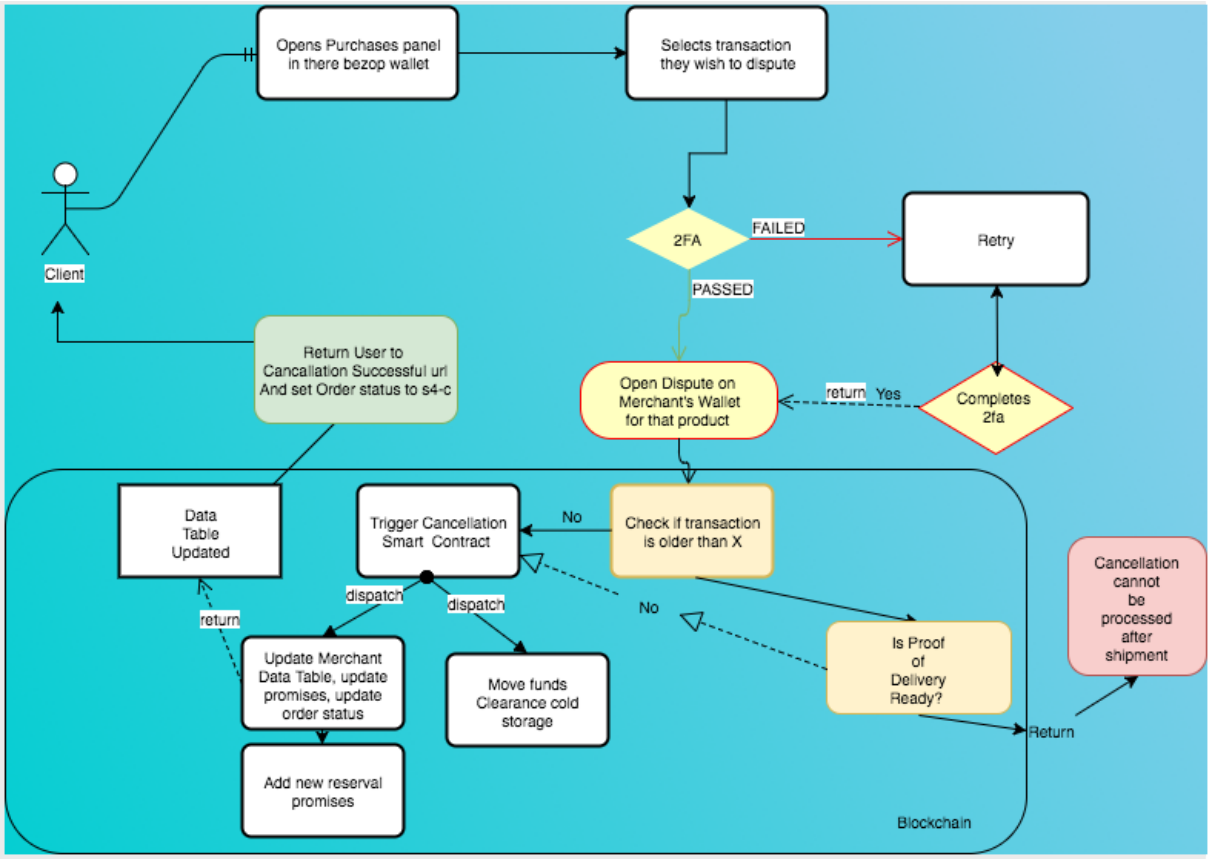
iii) The transaction was not cancelled

iv) i,ii,iii are met



Forced Order Cancellation (Smart Contract)

During order cancellation tokens are not sent the client until a safe period has elapsed ,a promise is stored on the merchant's transaction book to return funds to sender after clearance period, all previous promises are voided, the buyer is informed they will receive a refund after clearance days.

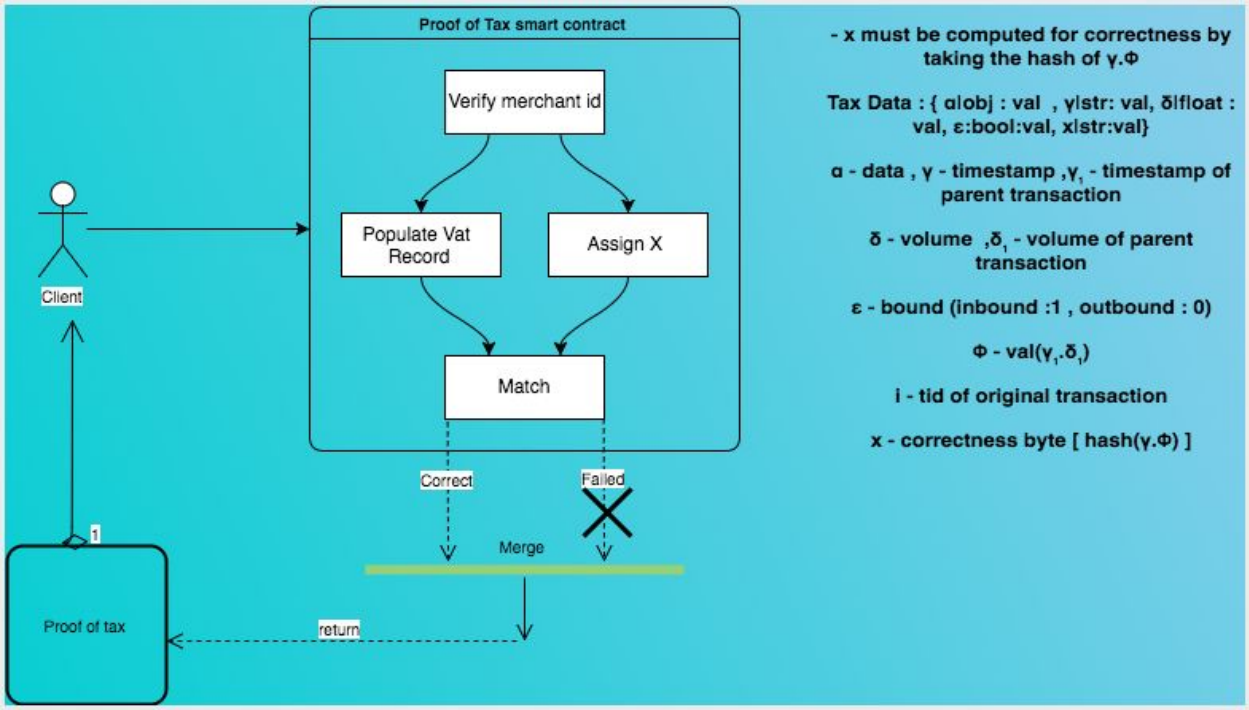


Note : within X days the buyer can cancel the transaction and reverse all payments made if the order is not shipped or they wish not to continue.

Cancelled funds are left in cold storage for 14 days before being restored to the buyer. Gas is required to trigger a cancellation smart contract.

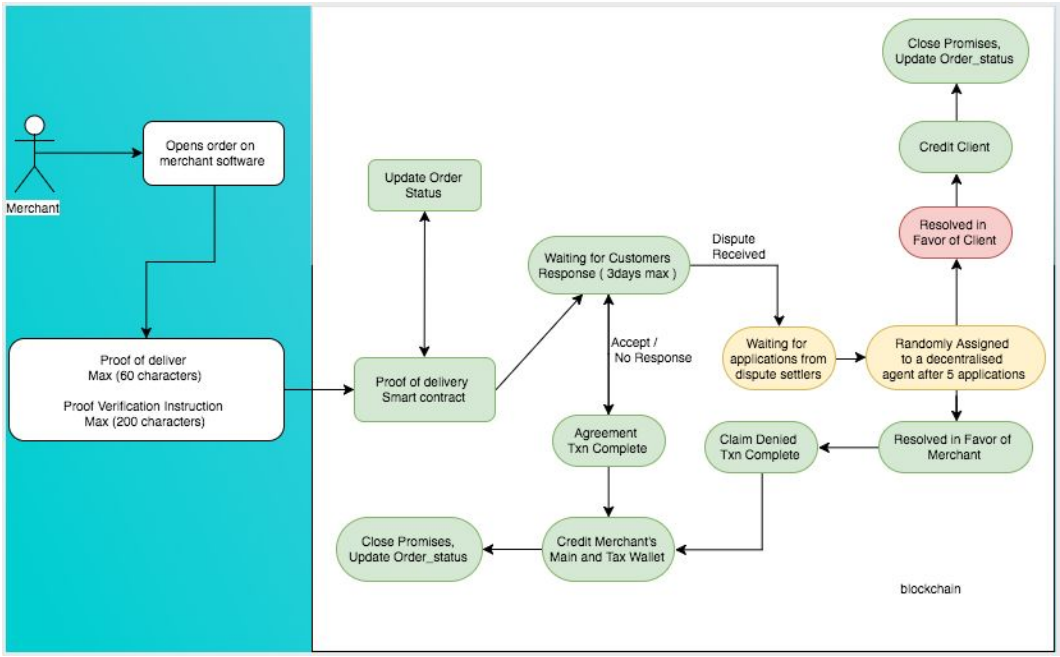
Proof Of Tax (smart contract)

This contract returns all tax data related to a merchant address



Proof Of Delivery (smart contract)

This contract submits proof of delivery for an order to the network.



Once proof of delivery is submitted, The order is marked delivered an extra safety layer allows the client 3 days to dispute the transaction if they have an issue.

A dispute smart contract is different from cancellation contract which is automatic. when a client files an issue with there order after X days have passed and merchant have provided proof of delivery. The network involves 3rd party miners who can earn 0.5% of the transaction by resolving the dispute.

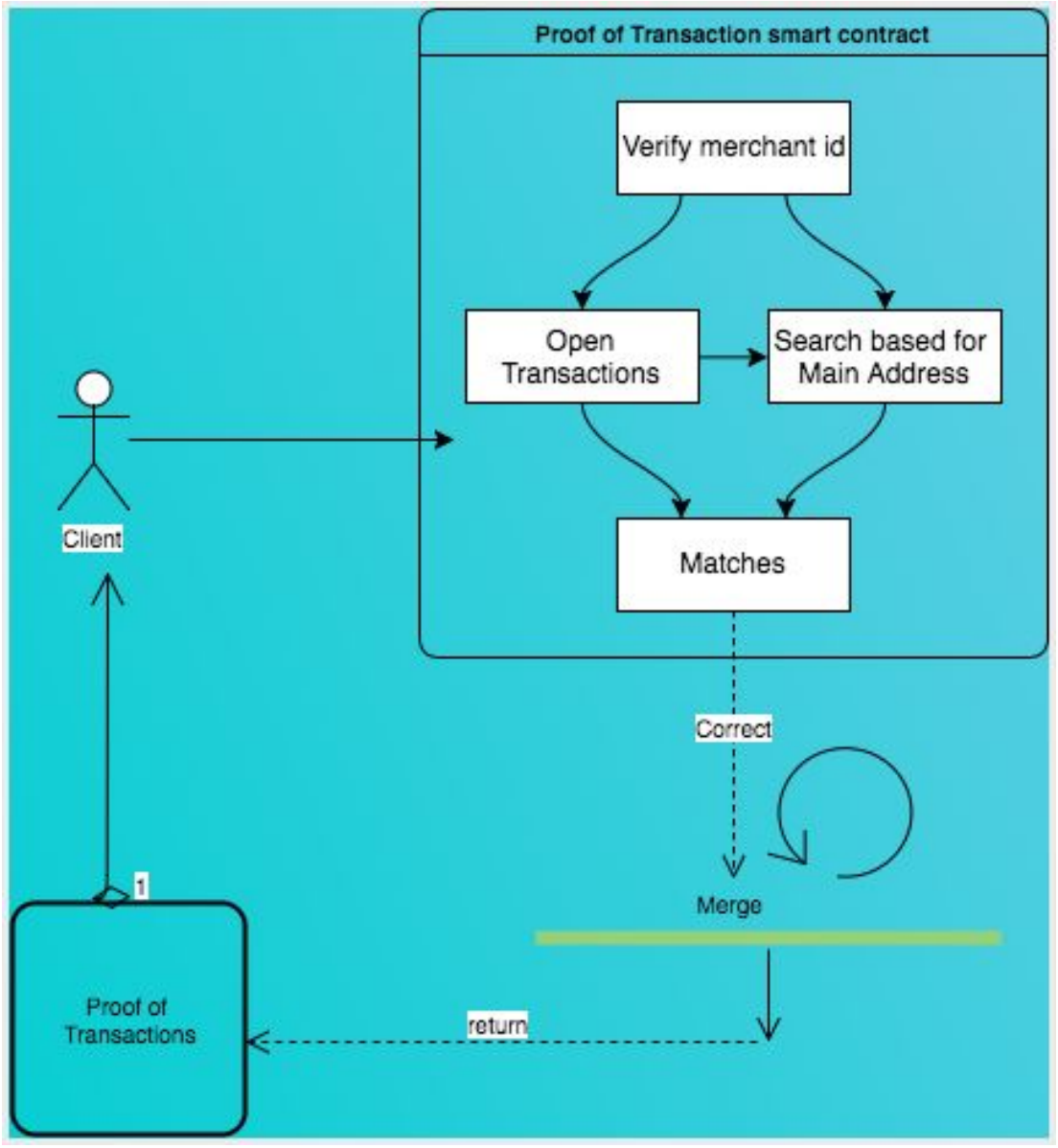
Creating a dispute requires gas.

Decentralized Dispute Agent

dispute agents can verify the proof of delivery provided by the merchant using the instructions and decide to resolve in favor of the client or merchant. Further input is not needed by the merchant or client.

Proof Of Transaction (smart contract)

This smart contract returns transaction data component , For an input merchant address.



4 Future Work

This work presents a clear and cohesive path toward the construction of the Bezop network; however, we also consider this work to be a starting point for future research on decentralized ecommerce systems. In this section we identify and populate three categories of future work. This includes work that has been completed and merely awaits description and publication, open questions for improving the current protocols, and formalization of the protocol.

4.1 On-going Work

Shapeshift Integration
Rainbow wallet
Bezop DOM
Smart Contracts
Bezop Dom Wallet Integration
Product Addition To Network

4.2 Open Questions

Will Bezop feature product rating ?

Yes , future smart contracts will allow users to add products into the network and all purchases must be be related to products on the network.

Post purchase , a client will have an extra option to post a rating after proof of delivery is submitted.

Will Bezop split off to its own network ?

Bezop is such a massive concept that a split out of the etheruem network is imminent

Once we have a stable product.

See the whitepaper roadmap - http://bezop.org/whitepaper.pdf

Does Bezop host stores for merchants?

At the moment , no .

In the future we may consider that as blockchain becomes more scalable

Merchants will have to find cheap and reliable solutions to host there stores.

Integration with other systems ?

We are at the moment researching this topic, Our Next paper will provide details about

This work.

4.3 Proofs and Formal Verification

Because of the clear value of proofs and formal verification, we plan to prove many properties of the Bezop network and develop formally verified protocol specifications in the coming months and years. A few proofs are in progress and more in mind. But it will be hard, long-term work to prove many properties of Bezop (such as scaling, offline).

References

[1] Vitalik Buterin. Ethereum , April 2014. URL https://ethereum.org/.

[2] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008. https://bicoin.org

[3] Wikipedia : Proof of delivery, *January 2012*: https://en.bitcoin.it/