Chapter 5
Ethereum, Smart Contracts, DApps

William Metcalfe

On February 28, 2012, an 18-year-old high school student wrote "If Bitcoin is to achieve mainstream success, it cannot stop at the limited crowds of Internet geeks, libertarians, and privacy advocates that it is hitting now, and it must find some way to attract the mainstream public (Buterin 2012)." At the time, 1 Bitcoin was worth 4.87 USD or 400 JPY. The Initial Coin Offering (ICO) did not exist. Ethereum was not yet even a proposal.

Whether you consider Bitcoin to be mainstream might depend on where you fall on the adoption curve, but the fact that you are reading this now means Bitcoin has most certainly reached beyond the "limited crowds of Internet geeks." The author of that quote was Vitalik Buterin in one of his early articles for *Bitcoin Magazine*. We will look at his role and contributions shortly. First, however, we consider how we arrived at the world of blockchain and DApps, starting with the concept of decentralization.

1 A Brief History of Decentralization

Decentralization is a fundamental part of the clever solution that gave us Bitcoin, the first widely successful digital currency. For currencies we know well, like Japanese Yen or US Dollars, or even other types of "currencies" such as customer loyalty points or air miles, we rely on a single authority like a central bank, an issuing

This article has been prepared for the study group "Blockchain and Society 5.0—The Creation of a New Marketplace based on Distributed Consensus" at the Research Institute of Economy, Trade, and Industry (RIETI) . I would like to thank RIETI and the study group for including me in this project as well as Makoto Yano, Chris Dai, and Jeff Wentworth for their gracious support and input throughout the process.

W. Metcalfe (✉)
Curvegrid, Tokyo, Japan
e-mail: william@curvegrid.com

M. Yano et al. (eds.), *Blockchain and Crypt Currency*,
Economics, Law, and Institutions in Asia Pacific,
https://doi.org/10.1007/978-981-15-3376-1_5

company, or another trusted custodian to guarantee the value of our money. Bitcoin does away with the need for a central authority by dividing the responsibility of protecting the network amongst the participants. But decentralization is as old as currency itself.

Gold used to be used as a decentralized currency because it too can be used without referring to a central authority. Someone can trade gold for goods and services with both parties recognizing its value. It has utility as a compact and fungible store of that value. Because of the similarity in concept, many describe Bitcoin as "digital gold" and Satoshi Nakamoto's famous paper uses the metaphor of mining, just like gold, to describe the creation of new coins (Nakamoto 2009).[1]

Of course, decentralization does not belong only to currencies. Modern Western democracy, created in ancient Athens and developed through the French and American revolutions, is practiced by the world's most advanced economies. It is our most familiar form of decentralization.

Computation also uses the pattern of decentralization outside of blockchain. NASA, for example, designed its space flight computers to be redundant by allowing multiple systems to vote on the output of a computation (National Aeronautics and Space Administration 1971). If one makes a mistake, the other computers will override it by majority.

It took, however, quite some time for decentralization and currency to rejoin forces since the retirement of gold as a day-to-day medium of exchange. The Knights Templar, a medieval Catholic military order, are often credited with inventing modern banking (Harford 2017). Twelfth century pilgrims would deposit their valuables with the Templars and receive a paper letter indicating the entitled value. Carrying the *promise* of gold was much more efficient and secure than carrying actual gold. Those promises are not too different from the currencies and instruments we use today.

Blockchain brings another type of efficiency to those promises by obviating the need for the Templars (or any other third party) by purely using technology. The first to use that technology successfully was Bitcoin. It allows a simple peer-to-peer value exchange from one account to another.

2 Ethereum

Vitalik Buterin became interested in Bitcoin at the encouragement of his father. After researching Bitcoin, he began writing articles in exchange for the cryptocurrency and started *Bitcoin Magazine* with another colleague. Eventually he had the revelation that the platform could become very powerful by being generalized beyond simple currency exchange into something that could perform *any* type of processing.

[1] Satoshi Nakamoto is well known to be an alias and the identity of the real author is unknown. L.S., "Who is Satoshi Nakamoto?", 2015, https://www.economist.com/the-economist-explains/2015/11/02/who-is-satoshi-nakamoto.

It may be easy to think of Bitcoin as a computer network that replaces your bank. But it is a little trickier to imagine how adding complex processing to Bitcoin could be useful. So let us approach it from the opposite angle and imagine your bank as a type of computer. It has three instructions: deposit money to my account, withdraw money from my account, send money from my account to another account.

Now imagine if you could give your bank special instructions to accomplish your savings goals: "for the next year, only allow me to withdraw up to 100 dollars per week." Or suppose you wanted to create a shared account for your startup business where the CEO has full control but wants an extra level of accountability for the other officers. "Make an account with three owners. Alice can withdraw as much as she wants anytime but Bob and Charles can only withdraw funds if one of the others also approves."

You could even automate the distribution of proceeds from your business "every time 20 dollars or more is deposited to the account, give 5 each to Alice, Bob, and Charles, and divide the remainder evenly amongst all other accounts on a special list." Each of those accounts might have its own special instructions! Maybe Bob wants all the funds to go directly to his favorite charity.

These are simple examples but something that would be really difficult to achieve with an actual bank because of the number of humans and processes involved; they are not equipped to provide that level of customization. It would involve power of attorney with someone you really trust, a series of elaborate legal contracts and independent bookkeeping, or all of the above. With a few lines of software, those examples can be created and the pattern replicated to anyone else who wishes to accomplish the same outcome.

It was possible that Bitcoin could evolve into this limitless computer. The developers with whom Buterin was working, however, were not receptive to this grand idea so he decided to embark upon the project himself. Thus, Ethereum was conceived around 2014 and launched in 2015 to extend the concept of Bitcoin. Ethereum has its own currency Ether (ETH) , just like Bitcoin (BTC), but the platform can run any set of instructions, not just "send and receive Bitcoin (Hacket 2016)."

2.1 Smart Contracts

Back in 1994, Nick Szabo, a computer scientist and legal scholar, created the term "smart contract" and defined it as: "A smart contract is a computerized transaction protocol that executes the terms of a contract (Szabo 1994)." He envisioned a way of bringing efficiency to written agreements in a way that enforces them automatically. Think of a vending machine. Without a shop clerk, it enforces the contract of selling a beverage at an advertised price to the customer who inserts a sufficient amount of money into the machine.

The Turing-complete (Wikipedia contributors 2019) computer engine provided by Ethereum is the first computerized transaction protocol that does many of the things Szabo was envisioning in his earlier writings. Computer programs that are

run on the Ethereum platform are called smart contracts. They can enforce *certain types* of agreements between parties, just like a vending machine, but they have no intrinsically direct relationship with legal contracts. The Ethereum smart contract became so popular that the Ethereum version of a smart contract has eclipsed the original use of the term and added a lot of confusion as to what blockchain can do.

Why is it a "contract"? The original idea was that it constituted some kind of agreement between parties. Why is it "smart"? The original idea was that it could execute itself without the need for lawyers or people to be involved. So what is a smart contract really? Since Ethereum declared itself "a decentralized platform that runs smart contracts," it really just refers to a special type of software program. It may or may not have legal implications and still needs a traditional legal framework around it if it needs to be used as part of a legal transaction. For example, if you write a smart contract to securitize real estate, dividing ownership of a property up into virtual tokens, you still need traditional legal contracts (in the appropriate jurisdictions) to tie those smart contract tokens to the actual property.

3 What Is a DApp?

What is it that emerges from the ability to manipulate numbers and data in a trustless manner provided by smart contracts and what would we call such an application? We can now decouple the application from an individual company or owner and create a "decentralized application" also known by the contraction DApp. It can be pronounced with two syllables as "dee-app" or with one syllable as "dapp." Capitalization is not always consistent (as in Dapp) and the D is occasionally written with Ð (as in Ðapp), where Ð is the Norse letter eth (ETH News).

Decentralized applications are often described as trustless or peer-to-peer with the distinguishing characteristic that there is no single server or entity controlling it like in a client–server model. We understand the attractive properties of the smart contract and the flexibility of the platform. But to understand what really makes a DApp different from a centralized application, it is worthwhile considering what goes into a contemporary centralized application.

A prototypical modern software application includes at least one user interface (UI); this could be a mobile app downloaded from an app store, a website (accessed from a computer or mobile device), or a desktop application installed on a computer. It usually involves data. This data could be provided by a single group or company, like a weather app using a national weather organization, or like in a social networking app it could be provided by the end users themselves. Finally, it involves some sort of manipulation of the data or computation.

A DApp uses the blockchain at the core of its data storage and processing. This is implemented using a smart contract. Currently the UI for a DApp is usually created using a traditional website model. So one can think of a complete DApp as a website

plus one or more smart contracts. A DApp has the same general properties as a traditional application. The main difference, therefore, is that the data and computation are provided by the blockchain.

3.1 DApp and Blockchain

The merit of using blockchains for DApps are as follows:

1. A user can see what is going to happen before executing a function or submitting any data.
2. Once the user has performed an interaction, it cannot be withdrawn, tampered with, or deleted.

By themselves these properties are useful. This embodies decentralization at a protocol level. However, this facilitates another type of decentralization that is a driving philosophical motivation behind DApps:

3. Governance can be decentralized so that the users of the application participate directly in its management.

   At this point, we consider two examples—one makes use of the first two properties, and another that helps demonstrate the idea of governance, or structural, decentralization.

CryptoKitties is one of the more famous decentralized applications (Bowles 2017). It is a game created by Axiom Zen that allows players to trade, breed, collect, and sell virtual cats. Unique or scarce tradeable items are a well-used pattern in gaming, both traditional and digital; however, in CryptoKitties, the virtual items are recorded on the blockchain. In this way, the actions are transparent and guaranteed, but there is nothing particularly decentralized about the ethos of the application.

Another example of a DApp is the DAO (decentralized autonomous organization) (Securities and Exchange Commission 2017). The application was governed by tradeable tokens that had voting rights. In this sense, the mechanism of the application was guaranteed by the decentralized Ethereum layer, but the concept itself was designed for decentralization of authority.

3.2 Coins, Tokens, and DApps

The terms “coin”, “token” , “cryptocurrency”, “virtual currency” , “digital currency” and, more recently, “crypto asset” are now frequently used in similar or interchangeable ways. We could continue to generate similarly exotic terminology by pairing different “crypto” adjectives with different “coin” synonyms! It is enough to confound even the most diligent financial linguist. Therefore, it may be quite some time before we all agree on the correct language to use, both casually and legally. For

now, however, we take a practical approach. We can break apart the different types of technologies and categorize them by the most accepted terms even though there may be some overlap in real world usage.

The primary distinction we will make is between "coins" and "tokens." We can describe coins as a base currency. When a network, such as Bitcoin, includes the currency as an integral part of the software, we think of that currency, in this case Bitcoin, as a coin. As we know, Bitcoin was created for the primary purpose of storing and exchanging funds. In Ethereum, the currency Ether is also built into the platform and is therefore a coin. This applies to other platforms derived from Bitcoin such as Litecoin or Monero. Coins are also used to incentivize good behavior and secure the platform. Coins are used to pay for computation and storage resources and are given to mining nodes as rewards for their work.

We can describe tokens as the units built on top of one of these base networks as a secondary feature. They are a way to take advantage of a robust and established blockchain network to create new digital assets. There is no need to convince users to join a new network or run new software. The token runs on and is secured by an existing network.

One of the earliest attempts to do this was with "colored coins" on Bitcoin (Bradbury 2013). Think of it like taking a poker chip and marking it with a special red stamp. It is still difficult to make a forgery, but now you can use it for another purpose, like a coupon for a free bowl of ramen. Mastercoin (which became Omni) was another attempt to extend Bitcoin by storing extra data along with the native Bitcoin transfer transactions (Buterin 2013). These were both creative attempts to leverage the technology but have some inherent difficulties in aligning with a platform that has its own independent design goals. Bitcoin, for example, introduced an upgrade to reduce the number of tiny transactions. This was done to prevent malicious degradation of the network, however, colored coins are optimized to minimize cost by making use of such tiny transactions.

Ethereum, having been designed from the ground up as a platform for running arbitrary computer programs, lent itself naturally to the creation of tokens on top of its platform. Using an Ethereum smart contract, a software developer can (comparatively) easily create a token with any amount of supply, distribution goals, or custom logic. The Ethereum smart contract formed the basis for most of what we call tokens today.

Tokens, like coins, do not have any innate properties besides their bookkeeping ability. However, they have a few basic genres under which they commonly fall. You may have heard the terms "utility token" or "security token." The reason this distinction is made has to do with how the tokens are used to raise capital for a project or business. Anyone can create tokens from nothing and sell them. The way Ethereum raised money for its development was by selling the platform's future currency (Ether) for Bitcoin. The concept was that Ether could be used to pay for the submission, storage, and execution of smart contracts and related data. It was this "utility" that would make it valuable in the future and a worthwhile investment. It could, of course, also be used simply as a medium of exchange like Bitcoin.

Inspired by the success of Ethereum, many other projects raised money by selling their own tokens. The sale of fractional ownership in a project to the public to raise funds, in exchange for a promise of a share of the future profits, is an economic practice dating back hundreds of years. The potential for defrauding or disappointing investors by lying about a project's potential, absconding with the money, or simply failing to execute, means we have sophisticated laws and regulations in economically advanced societies to prevent good people from being duped into funding bad projects.

Selling cryptocurrency-based tokens proved to be an attractive way to raise a lot of money compared with traditional financing. ICO funding reached a peak in the first quarter of 2018 where blockchain startups raised an astronomical $6.9 billion through ICOs compared with $0.5 billion through equity funding (CB Insights 2019). To avoid running afoul of existing securities laws, many projects took great pains to indicate the tokens they were selling were utility tokens and therefore not subject to existing regulations. There are a number of conflicting legal opinions on the subject. Whether the regulators eventually decide there is a place for utility tokens (and what that place is), the markets as of mid-2019 have had their fill. The span of 2017–2018 was an exceptional period and there is effectively no current interest in such projects (Vigna 2019).

But what about security tokens? A traditional security means attaching a paper or digital legal agreement to a physical object or to a company or project. This could be shares in a company like Toyota or a government bond that pays interest. These types of instruments are things that could be easily modelled or represented in the blockchain. If we allow tokens to represent securities, we can draw on the advantages of tokens, such as broad access to capital with low management overhead, with the reliability and responsibility of legally regulated instruments.

3.3 The Case for DApps

There are a number of applications being pursued in the DApp space. So far, they mostly fall under the following general areas:

- Fundraising (ICOs)
- Marketplaces including exchanges
- Identify providers—know your customer (KYC) and anti-money laundering (AML)
- Financial services
- Securitization of assets
- Supply chain management
- Gaming.

We now examine some of these applications.

To provide financial or securities services like a bank or stock exchange in a pre-DApp world, the barrier to entry is significant. The burdens of regulatory compliance,

staff, infrastructure, and institutional relationships required to operate are staggering. In the United States, the estimates to start a bank, for example, are $12 to $20 million in capital (Harrington 2016).

Using a smart contract-based system, where the deposits are governed by publicly visible computer code, anyone with the ability to write software can create a system that securely handles large amounts of assets. An ambitious software developer in 2016 created a working cryptocurrency exchange called EtherDelta (Winters 2016). The exchange smart contract held over a billion dollars of ETH and tokens at its peak.

One of the frequently cited goals of decentralized projects is "self-ownership" of data. After so many security breaches and revelations of large companies selling the personal information of users, DApps provide a chance for users to have more control over their data.

A well-designed smart contract system could allow regulators or lawmakers to authorize or monitor certain activity on a platform. Imagine a sales program where merchants and consumers could register in a marketplace. Each participant would have a unique identity in the program and transactions could be posted and settled directly through the program. Although it might sound undesirable to some less than scrupulous audience members, every transaction could be automatically taxed to the appropriate level by the government. When tax rates change, the government could simply adjust the rate in the smart contract directly and there would be no onerous actions required by merchants to implement the new rates. Moving control to a common system instead of disparate bureaucratic entities has great potential to align societal and commercial interests.

4 Where *Is* the Smart Contract?

Blockchain transactions are stored on a computer, commonly referred to as a node. Popular blockchain platforms such as Ethereum have tens of thousands of nodes operating at any given time. Each node stores an identical copy of all of the transaction records. The node that validates the next batch of transactions, referred to as a block, is called a mining node. The blocks are in turn validated by each mining node.

Every node in the Ethereum network stores a copy of all the software (smart contracts) , data, account balances, and transaction state (Buterin 2014a). If you think that sounds like a lot of data, it is. A copy of the production Ethereum node at the time of writing is about 179 GB of data. A full archive of all transactions that have occurred, including all intermediary states is 1.8 TB.[2] For comparison, a typical smartphone or laptop might have 64–512 GB of total storage.

Transactions are transmitted across the Internet between nodes in a peer-to-peer fashion. That is, nodes have connectivity with some but not all of the nodes in the network. It takes about 40 s for a given transaction to be seen by 95% of the nodes

[2]Running geth version 1.8.18-stable on Ubuntu Linux. https://geth.ethereum.org/.

in the comparable Bitcoin blockchain (Decker and Wattenhofer 2013). A transaction fee is submitted alongside the transaction and the mining node receives the fees for the transactions it groups into the block.

The mining refers to guessing the solution to a mathematical problem (unique to the group of transactions) that cannot be calculated directly. That guessing is done on hardware that can make millions of guesses per second. There is an expected average number of guesses it will take to get to the solution and so the mining node has effectively proven that it has executed a number of guesses. Hence mining is also known as proof-of-work (PoW).

The mined block is distributed back to the network over the Internet and each node will verify all the contained transactions before accepting it and passing it along. Eventually all the nodes will store a copy of the system state that they all agree upon. In this way, it is not so much a distributed computer like one would think of in a traditional sense. It is more like one computer with many clones running in parallel and storing the same information to make sure no one cheats. This is similar to the redundancy of the aforementioned NASA space flight computers but on a much larger scale.

5 DApp Development and Challenges

Ethereum has been around since 2015 and we have only seen a few years of development in the ecosystem. Compare this with some of the early technologies of the Internet. NCSA Mosaic, launched in 1993, was the first graphical browser to popularize the World Wide Web. This was followed by the PHP language, MySQL database, and Apache web server in 1995. These types of tools allowed early technology pioneers such as Amazon and Match.com to create useful applications.

Blogging platforms Blogger, Movable Type, and Wordpress were launched in 1999, 2001, and 2003, respectively. These made it possible for more enthusiasts to participate in the Internet. But it was not until services like Facebook became available and popular that the Internet felt broadly participatory.

5.1 How Are DApps Made?

Many of the tools used to produce and consume DApps are still in their infancy. On the user side of interacting with a DApp, you need a way to create and manage an account on the network. In a traditional application, login information (email and password) is stored on a server. In a DApp, your account is a digital blockchain key stored on your computer's drive or in your smart phone's memory. There are tools to help you manage those keys. The most popular tool to manage DApp (Ethereum)

accounts and interact with DApps is MetaMask,[3] an extension for the Chrome, Firefox, and Opera web browsers.

Unlike vanilla web browsers, which automatically upgrade and are relatively stable pieces of software, the new DApp browsers and plugins are often buggy. Furthermore, the new interaction model with DApps including icons, jargon, and actions are still confusing to new users.

There are also popular web-based wallets, such as MyEtherWallet,[4] which allow you to conduct transactions without installing any software. However, it is geared primarily to exchanging Ethereum as a currency and not interacting with DApps. Interacting with a DApp requires you to copy and paste arcane computer code into a web form.

Hardware wallets, such as Trezor[5] and Ledger,[6] are a third type of wallet. They store the encryption keys in a tamper-proof module from which the digital keys cannot be physically removed. That way a user needs to physically connect a device and approve an action. The challenge here is that extra work can be required to set up, understand, and use the device. For DApp and software developers, integrating hardware wallets requires extra development, testing, and consideration.

The aforementioned software and hardware are primarily for end users but are used extensively by developers during the construction of a DApp. There is another suite of tools used only by software developers, including integrated development environments (IDEs) such as Remix, testing frameworks such as Truffle,[7] and the main programming language for Ethereum called Solidity.

Automated testing is another critical component in development. Truffle and its complementary tools Ganache and Drizzle are the main tools used for testing. They let you connect and deploy to a simulated Ethereum network on your own computer to put the smart contract through its paces.

The Ethereum community maintains a series of public test networks as well. In the final stages of development, your smart contract can be deployed to one of these networks for a fully decentralized run-through. Truffle and its components are young like the rest of the toolset and the execution of tests can take more time and effort to run compared with more mature web development frameworks. However, for reasons explained below, thorough testing is an even more critical part of the development cycle when compared with most web applications.

[3] https://metamask.io/.

[4] https://www.myetherwallet.com/.

[5] https://trezor.io/.

[6] https://www.ledger.com/.

[7] https://www.trufflframework.com/.

5.2 Smart Contract Maintenance

In the early days of personal computers, most version upgrades took place by purchasing the latest copy of a title and manually installing it. Today, much of the software we use exists as an online web application and is upgraded instantly and transparently by the owner of the website. It is a frequent pattern with contemporary desktop and mobile phone software to enable automatic upgrades. Chrome browser, for example, updates itself by default.

The smart contract upgrade model is unlike either of these models. Ethereum smart contracts are, by design, immutable once deployed. This implies a number of considerations that are very different, even contradictory, to the prevailing philosophies of web development. Consider Facebook's (now retired) motto "Move Fast and Break Things." For the world of smart contracts, one might propose the motto "Move Carefully and Test Thoroughly so Things Never Break."

Much of the community and audience for DApps comes from a web centric background where certain types of rigor have lost favor and been replaced with rapid iteration and disposability. Testing has a prominent role in smart contract development. Besides testing, the rather academic discipline of formal verification has made its way into the blockchain discourse. Techniques used in industrial, mass transit, aerospace, and other fields where mistakes have huge consequences can also be applied. But systems will still need to grow, and, despite the best intentions, mistakes will still be made.

Given that we expect and plan for platforms to evolve, there are a few ways, at least in Ethereum, that one can approach a path for upgrades.

1. The deployed smart contract (contract A) can be a pointer to another smart contract (contract B) that implements the actual functionality. Somewhat like a mail forwarding address. If the functionality needs changing, A's reference to contract B gets updated to point to a replacement (contract C).
2. The smart contract uses replaceable underlying libraries to implement its functionality. This is conceptually similar to the first technique and differs mainly in technical nuances.

Both of these methods allow for a seamless transition to the new system (provided there are no problems or compatibility issues), but it explicitly removes one of the properties that makes smart contracts valuable. You do not know what will happen in the future. In a basic token implementation, will someone rewrite the underlying smart contract so that my assets can now be garnished by a party I may not trust? If it is immutable and non-upgradeable, you can verify with some confidence that your tokens will be secure.

However, there is one more upgrade method that can work even if the deployed smart contract is completely immutable and non-upgradeable.

3. You make a new smart contract and tell everyone to use it instead of the old one.

Take an example of a token backed by copper. The fictional company "Acme Copper Coins" buys a bunch of copper and puts it in a warehouse somewhere. They create

a smart contract to issue tokens against their copper supply. But perhaps they underestimated the demand for their tokens and the smart contract was designed to only support up to 30,000 coin buyers.

Acme could make a new smart contract that supports up to 60,000 buyers and declare they are copying your token balances from the old smart contract to the new smart contract. From now on they are no longer going to honor redemptions of the old tokens. This is ostensibly OK because my new token is worth the same as my old token. However, there is a limit to what a blockchain system can directly control. Beyond that limit, we still rely on the instruments and conventions that exist in our present society such as traditional contract law, public reputation, and trust. We still have to trust Acme that they actually have copper in their warehouse and that I can trade in my tokens with them for copper if I want to. Moreover, we might expect the traditional justice system to intervene if they renege on that promise.

6 The Boundary Between DApps and the Real World

DApps are still a developing technology. To be put into wide use, it is necessary to overcome various issues, including consistency with regulations, data reliability, and the ability to respond to expanding demand (scalability) .

6.1 Consistency with Laws

Legal and regulatory frameworks are an important consideration when developing many applications that hope to migrate to the blockchain. Take real estate tokenization for example. If Alice sends a token to Bob that represents a share in a particular piece of property in Tokyo, then that transaction can take place in a way that is guaranteed and verifiable by the technology. Alice could set a price and Bob can see that if he pays that price, the token will reach his custody. Nothing can prevent Alice from withholding the token or Bob from withholding payment. In this example, the blockchain replaces the escrow function of a trusted third party. It does not, however, replace the fact that there needs to be laws that tie the share of property to that digital token.

6.2 Reliability of Data

Getting trustable data into the blockchain is also an issue. Let us say you wanted to create an insurance scheme that would pay out if the temperature gets too cold. Perhaps farmers would pay into this and receive compensation if the temperature drops

below a specified threshold. You could create a system where anyone could participate as an underwriter and anybody could participate as a policy holder. Effectively parties would be taking opposite sides of a bet on the weather.

A system that feeds real-world data into the blockchain for use by smart contracts is known as an "oracle." By trusting a smart contract system that depends on an oracle, you are implicitly trusting that oracle as a reliable source of data. If you allow many oracles to provide data in a decentralized manner including incentives for telling the truth and disincentives for cheating, then you can create a more robust system.

Randomness is another piece of complexity in the blockchain worth mentioning. Randomness is used in many cryptographic systems and techniques to guarantee fairness. Because the internals of a smart contract and the participants' attempts to interact with it are visible to the blockchain network, a miner could gain an advantage by knowing or altering the outcome of a transaction. For example, if a smart contract-based lottery for highly coveted tickets to a sporting event relied on seemingly unpredictable data such as the block creation time, the block miner could adjust the publication time to manipulate the result.

6.3 Scalability

Like Bitcoin, Ethereum grew organically as an experiment. As more people join the network, the demands on the technology become higher. Scalability is the potential of the system to meet those growing demands. To frame the topics of scalability it helps to understand the fee structure surrounding transactions and block creation.

Currently the target block creation time, a compromise between security, efficiency, and practical network limitations, is 12 s between blocks (Buterin 2014). To maintain this rate, the mining difficulty is automatically adjusted by the Ethereum software as mining power is added or removed from the network.

As mentioned earlier, users of Ethereum submit a fee when submitting transactions to the network. This fee is measured in units called gas and relates to the size and complexity of the transaction. The miners claim a per-transaction fee as part of their incentive to secure the integrity of the blockchain and the fee structure helps balance the supply and demand for transaction processing.

There is a block gas limit agreed to by mining nodes, which caps the maximum amount of fees that can be accepted into a block. This is used to manage the bandwidth, cost of storage, and cost of computation per block. That gas limit is about 8,000,000 at the time of writing, and with an average transaction size of about 80,000 gas, about 100 transactions will fit in a block. Hence, the network can currently process about 8 transactions per second.[8]

Some of the main topics in scalability are:

- Transaction throughput

[8] https://etherscan.io/.

This is the quantity of transactions (currently 8 per second), such as sending and receiving tokens, that the blockchain network can process per unit of time. This number is often compared with Visa's 24,000 transactions per second (Visa, accessed 26 November 2019). However, we must remember that these systems were made for different purposes and even then, an Ethereum payment transaction is really like payment, clearing, and settlement all in one.

- Computational cost
  Block validation and mining are costly in terms of computer hardware, electricity, and ultimately fees paid by the users of the network as explained above. All the mining nodes in the network perform all the transaction computations and this is inefficient.
- Data storage
  In Ethereum, currently all full participants keep a complete record of all blockchain transactions. Because of the amount of accumulated data in the blockchain, it is untenable on most consumer devices now to run a full Ethereum node.

The areas where blockchain seems slower or less efficient than its nonblockchain counterpart technologies are caused by the trade-offs that enabled it to work in a decentralized manner. There is, of course, active work and research being done to improve real or perceived shortcomings and eliminate obstacles to growth.

Proof-of-stake (PoS) is an alternative to PoW mining systems that will allow a higher transaction rate. Rather than commit computing resources to guessing a mathematical answer to verify a block (as in mining) , users will be able to pledge Ethereum for a period of time to gain voting rights on block confirmation and receive a reward in exchange for helping to secure the integrity of the blockchain network. They cannot use this pledged Ethereum and they could forfeit their stake if they cheat.

The computation work and data storage can be split amongst different groups of nodes. That way both computation and storage can be divided with a sufficient level of redundancy. This technique is known as sharding.

Light nodes are another way that the network becomes more accommodating to different players. Mobile devices, for example, lack the storage capacity to participate in most public blockchains. A light node can contain the data required to perform minimal validation on transactions and act as a conduit to the broader network, relaying data both to and from the client.

6.4 The Future of DApps

We have been looking mostly at Ethereum; however, there are a number of competing technologies being developed of which Ethereum is just one. Not all of these technologies will proliferate. If we look at the late 1980s and early 1990s, Mintel was a popular network in France with similar models attempted around Europe and

other parts of the world, while CompuServe existed in the United States. Eventually, both of these services were fully supplanted by the Internet.

However, older technologies do not always disappear just because newer or better technologies emerge. Legacy technologies can continue to exist alongside newer ones. For example, even though voice and video applications such as Skype or Google Hangouts allow you to communicate over the Internet, the enduring telephone network shows no signs of disappearing. Not only do the networks coexist they even seamlessly integrate. You can make a phone call from Skype or dial into a Google Hangouts conference. Established PoW systems may continue to exist and even interoperate with newer PoS systems.

In the coming months and years of blockchain evolution, we expect some of these competing and overlapping systems to integrate. Polkadot,[9] for example, is a platform designed to aggregate and bridge multiple different blockchains and subnetworks. Bitcoin and Ether might trade with each other under a system of shared security (Parker 2019).

The UIs in contemporary DApps are still accessed in a centralized fashion. In one way this is OK in the philosophy of decentralization. The critical parts are decentralized on the blockchain. Ideally the whole application, including the images and visual UI components, are decentralized as well. The blockchain can be used to help secure and distribute those files, not just the programs and tokens. The InterPlanetary File System (IPFS)[10] is one example of a distributed data storage protocol that uses the blockchain to do that. In the future we may see DApps become fully decentralized by using protocols like IPFS to store and serve their files.

In the meantime, the platforms we know will continue to evolve. Proposals for changes to Ethereum can be submitted by anyone and each new proposal is numbered based on its order of submission. EIP-20, the 20th proposal (Vogelstellar and Buterin 2015), became the ERC-20 standard that helped facilitate the ICO boom. With ERC-20, token creators, token exchanges, and Ethereum wallet software could implement a common interface and all these different parties could work together making interoperable products. It is similar to how Sony and Philips released the CD Audio standard so that manufacturers of CDs and CD players, along with music producers, music stores, and consumers, could all use the same type of disc.[11] As of May 2019, the most recent proposal number for Ethereum was 2015.

Vitalik's early article in *Bitcoin Magazine* was exploring "If Bitcoin is to achieve mainstream success…". For centuries we have used money in nearly the same way it was invented—trading pieces of metal for goods and services. The public blockchain is only a few years old and mainstream *awareness* of blockchain continues to grow. We have already seen a number of highs and lows, but true mainstream *success* will be a long-term effort.

---

[9] https://polkadot.network/.

[10] https://ipfs.io/.

[11] Ethereum software, however, is free and open whereas the CD Audio standard was commercially licensed.

References

Bowles N (2017) CryptoKitties, explained … mostly. https://www.nytimes.com/2017/12/28/style/cryptokitties-want-a-blockchain-snuggle.html

Bradbury D (2013) Colored coins paint sophisticated future for Bitcoin. https://www.coindesk.com/colored-coins-paint-sophisticated-future-for-bitcoin

Buterin V (2012) Bitcoin adoption opportunity: teenagers. https://bitcoinmagazine.com/articles/bitcoin-adoption-opportunity-teenager-1330407280/

Buterin V (2013) Mastercoin: a second-generation protocol on the Bitcoin blockchain. https://bitcoinmagazine.com/articles/mastercoin-a-second-generation-protocol-on-the-bitcoin-blockchain-1383603310/

Buterin V (2014a) A next-generation smart contract and decentralized application platform. https://github.com/ethereum/wiki/wiki/White-Paper

Buterin V (2014b) Toward a 12-second block time. https://blog.ethereum.org/2014/07/11/toward-a-12-second-block-time/

CB Insights (2019) Blockchain trends in review. https://www.cbinsights.com/research/report/blockchain-trends-opportunities/

Decker C, Wattenhofer R (2013) Information propagation in the Bitcoin network. https://tik-old.ee.ethz.ch/file//49318d3f56c1d525aabf7fda78b23fc0/P2P2013_041.pdf

ETH News (n.d.) Definition of 'Dapp'. https://www.ethnews.com/glossary/dapp

Hacket R (2016) Can this 22-year-old coder out-bitcoin Bitcoin? http://fortune.com/ethereum-blockchain-vitalik-buterin/

Harford T (2017) The warrior monks who invented banking. https://www.bbc.com/news/business-38499883

Harrington R (2016) How to start your own bank. https://www.huffingtonpost.com/2010/03/19/how-to-start-your-own-ban_n_497261.html

Nakamoto S (2009) Bitcoin: a peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf

National Aeronautics and Space Administration (1971) Spaceborne digital computer systems. https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19710024203.pdf

Parker E (2019) Ethereum co-founder: 'my biggest disappointment is how toxic this space has become'. https://www.longhash.com/news/ethereum-cofounder-my-biggest-disappointment-is-how-toxic-this-space-has-become

Securities and Exchange Commission (2017) Report of investigation pursuant to Section 21(a) of the Securities Exchange Act of 1934: The DAO. https://www.sec.gov/litigation/investreport/34-81207.pdf

Szabo N (1994) Smart contracts. http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html

Vigna P (2019) Raising money in the Crypto World has gotten a lot harder. https://www.wsj.com/articles/raising-money-in-the-crypto-world-has-gotten-a-lot-harder-11554037201?mod=rsswn

Visa (2019) Visa acceptance for retailers. https://usa.visa.com/run-your-business/small-business-tools/retail.html. Accessed 26 Nov 2019

Vogelstellar F, Buterin V (2015) ERC-20 token standard. https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20.md

Wikipedia Contributors (2019) Turing completeness. https://en.wikipedia.org/wiki/Turing_completeness

Winters T (2016) Decentralized exchange on the Ethereum blockchain. https://www.ethnews.com/decentralized-exchange-on-the-ethereum-blockchain

William Metcalfe is a co-founder of Curvegrid, a blockchain software startup based in Tokyo. Previously, he was the chief technology officer of Gilt Japan after joining the organization in New York City as an early employee. Mr. Metcalfe helped grow Gilt from its first order to become one of the leading ecommerce sites. He graduated from the University of Waterloo with a degree in computer science.