# Survey of Blockchain Consensus Algorithm Scalability

Ryan West
ryanwest6@gmail.com
Brigham Young University

***Abstract* — Blockchain technology relies on consensus algorithms to propagate, validate, and confirm new transactions in a distributed fashion. However popular algorithms such as Proof of Work (PoW), used by Bitcoin, are inefficient, have long latency, and have a maximum transaction throughput. To increase blockchain scalability, a multitude of consensus algorithms have been proposed which alter the blockchain in various ways. This paper presents a survey of these consensus algorithms, focusing on how well they scale in terms of transaction throughput, latency, fault tolerance, and security.**

## 1 INTRODUCTION

Blockchain technology has been the focus of much research recently. An immutable distributed database of transactions, blockchain seeks to achieve decentralized consensus of blocks of transactions between untrusted groups without third party management. Many blockchain implementations have found wide success around the world such as Bitcoin, Ethereum, and Ripple. As the number of users per blockchain increases, several scalability problems surface which increase costs, lower performance, and often become the culprit for a business blockchain failure. While implementations vary widely, scalability problems such as low transaction throughput and confirmation latency persist across virtually all platforms and may require significant technical changes to improve performance.

As an example, Bitcoin is the most well-known blockchain to date with over 100 million account owners, 500 thousand unique addresses in transactions per day, and over $150 billion in value traded(1,2). Bitcoin uses the Proof of Work (PoW) consensus algorithm which requires participating nodes to compute an increasingly difficult cryptographic puzzle to add a new block of transactions to the chain. While Proof of Work allows for a completely decentralized, public blockchain, "miner" nodes competing to solve each puzzle now consume more power than all of Singapore [1]. Each block of transactions is limited to 1 MB in size and each block takes approximately 10 minutes to be added to the blockchain, yielding a transaction throughput of only 7 TPS (transactions per second). In contrast, major payment processors such as Visa typically process over 4000 TPS. Once a block of transactions is successfully added to the network, many Bitcoin payment processors require 6 additional blocks to be added to minimize the probability of a fork, in which two separate chains of transaction history emerge in the network. This adds an hour or more of latency to transaction processing time. High demand for Bitcoin compounded with limited throughput results in very high transaction fees, closing off the network for lower-value transactions. This pattern is difficult to sustain and has prompted the creation of many additional consensus algorithms meant to solve some of the problems faced by Bitcoin.

In this paper, we review many commonly used consensus algorithms and measure their performance, with an increased focus scalability. We first cover some of the major blockchain overheads, then categorize different types of proposed scalability solutions in the blockchain space. We then cover public, private, and consortium blockchains and define a few factors that help measure each algorithm's scalability. Finally, we categorize and discuss each consensus algorithm, first covering Proof-of-X algorithms and then Byzantine Fault Tolerant algorithms.

## 2 BACKGROUND

### 2.1 Major Blockchain Overheads

There are two major overheads incurred by blockchain technology which can be major obstacles when trying to expand the blockchain[16]:

(1) **The need to run a consensus protocol before state can be updated**. Being a distributed database, every honest node must agree which transactions are valid and accepted into the database. This is done via consensus protocols such as Proof of Work (PoW) or Practical Byzantine Fault Tolerance (PBFT)[5], in which nodes gossip information about transactions with each other and gradually obtain the same collective state. As previously seen with Bitcoin, some consensus protocols require significant computing power and lead to insufficient transaction throughput as number of users and number of nodes increase.
Many solutions have been proposed to lower transaction latency and throughput. These often include a modification to the consensus algorithm, using consortium or private blockchains, or other network optimizations. This research is collectively grouped into the network layer.

(2) **The need for full system provenance**. Full system provenance exists when a system can automatically determine how each piece of data came to existence [2]. Blockchains maintain full system provenance by storing the complete transaction history on every node, allowing each node to recalculate all transactions from blockchain inception to present day to verify that the data is and was always accurate. Bitcoin's transaction log now exceeds 260 GB at the time of writing [2]. While integral to ledger auditability, this requirement creates a costly storage burden for node operators who often cannot use commodity hardware to participate.
Research focused on decreasing storage size while maintaining auditability includes many data sharding and compression proposals, pruning old data, thin (small) clients and nodes, and providing better incentive systems to pay node operators to store data. This research comprises the storage layer.

[1] https://www.cbeci.org/comparisons/

[2] https://ycharts.com/indicators/bitcoin_blockchain_size

### 2.2 On-Chain, Cross-Chain, & Off-Chain Solutions

Existing solutions for improving blockchain scalability can be categorized into three types: On-chain, Cross-chain, and Off-chain. **On-chain** solutions modify the blockchain model itself, including its distributed consensus algorithm, ledger, storage, and other internal pieces. **Cross-chain** solutions are a step above this and may use multiple blockchains to spread out data storage and take advantage of geographic locality. **Off-chain** solutions are a step further removed and may include tracking groups of transactions outside of the blockchain entirely, or other more business-focused solutions.

While each type of scalability solution may prove to be effective, Cross-chain and Off-chain solutions are ultimately built on and influenced bycore blockchain technology. This survey focuses exclusively on on-chain scalability research and specifically on consensus algorithms.

### 2.3 Public, Private, & Consortium Blockchains

Blockchains can be categorized by use into three different types: public, consortium, and private.

A public blockchain is completely open to the public; anyone can anonymously write data to the ledger, read the entire ledger, and operate a node that participates in consensus. This is the most well-known type of blockchain and includes major players such as Bitcoin and Ethereum. A public blockchains has completely decentralized authority and is often chosen for this purpose, but it can be expensive to run, hard to incentive node operators, and does not scale well, resulting in low speeds.

A private blockchain is the opposite—they are accessible only internally to the company that runs it. While this allows the company to control costs and technical details, it defeats the purpose of having a decentralized, trustless database and is only used in specialized situations.

A consortium blockchain is both public and private—it typically allows the public to read and possibly write transactions to the ledger, but it restricts node ownership to a set of organizations. Authority is distributed between the node operators and leadership is often conducted through a council or committee. With a limited number of nodes participating in consensus, this allows the blockchain to use more secure consensus protocols such as Practical Byzantine Fault Tolerance (PBFT) and achieve higher speeds.

## 3 SCALABILITY

### 3.1 What is Scalability?

Scalability is the unique characteristic of how well a system can be expanded to handle increases in load. It measures how well the system maintains the same performance in terms of job execution time and latency, cpu time, memory usage, Disk and Network I/O, number of errors and crashes, and many more factors. It is important to recognize that scalability measures performance as load increases—a single-server application with excellent response time for a few users is less scalable than a multi-server cluster application with a steady response time for thousands of users.

There are two main ways of scaling, known as vertical scaling and horizontal scaling. Vertical scaling equates to adding more power to an existing resource, such as increases in RAM and CPU power or faster/more storage such as high-capacity Solid State Drives. Horizontal scaling adds more servers to the system and distributes load across each of them. Vertical scaling is often considered simpler as it is not difficult to upgrade one system's hardware. However, horizontal scaling is often necessary for large-scale projects, whose load surpasses the abilities of one machine or is geographically distributed and requires low-latency local endpoints. Different blockchain projects may focus on horizontal or vertical scalability.

One interesting difference between traditional scalability discussions and decentralized ledgers is horizontal scalability. Adding more servers to a cluster typically means increasing load capacity; however, since each blockchain node verifies the same transactions, adding more nodes usually does not increase this capacity. In fact, it often *decreases* load capacity because of the additional network communications required to support these new nodes in the network.

### 3.2 Blockchain Scalability Factors

Scalability applied to blockchain often focuses on how well the blockchain can handle increasing rates of new transactions on the network, maintaining network speed, or keeping the network reliable and secure as load increases. The term is often used vaguely and in order so quantify this study, we measure a number of discrete factors.

(1) Number of Nodes. Each node that participates in consensus further decentralizes authority and trust; however, it must also validate all transactions and communicate with other nodes. How many nodes can the blockchain tolerate without significant drops in performance?
(2) Latency. How long does it take for a new transaction, block, or vote to propagate across the network?
(3) Transaction Throughput. How many transactions can be processed per minute, hour, or day?
(4) Transaction Confirmation Latency - How long must someone wait before they can be sure their transaction is final (i.e. isn't susceptible to a fork)?
(5) Fault Tolerance. How much of the network can partition or misbehave (malicious or accidental) without affecting overall network performance?

We also consider the security and privacy that each algorithm offers and note where trade-offs may be made.

## 4 CONSENSUS ALGORITHMS

A consensus algorithm or consensus protocol distributes, validates, and confirms new blocks of transactions throughout all nodes running the blockchain. There are many different consensus algorithms, each with its own advantages and disadvantages. This sections analyzes several of these algorithms with a focus on performance and scalability. These mechanisms are categorized into Proof-of-X Algorithms, which are generally best suited for large, public blockchains, and Byzantine Fault Tolerant Algorithms, which offer increased fault tolerant but function best with consortium or private blockchains.

## 4.1 Proof-of-X Algorithms

As Bitcoin is commonly considered the pioneer of blockchains, its consensus algorithm, Proof of Work, has received lots of attention. A class of related algorithms has emerged that are based on Proof of Work but are instead named Proof of *X*, varying the final word. These algorithms typically attempt to prove that transactions are cryptographically verifiable and valid using a variety of techniques. They tend to be designed for public, permissionless networks in which anyone can read and write to the blockchain.

*4.1.1 Proof of Work.* **Proof of Work (PoW)**, occasionally referenced as Nakamoto consensus, is the original consensus algorithm proposed by Satoshi Nakamoto in the first Bitcoin white paper[15]. When a user submits a transaction, the receiving node sends this transaction to other nodes and gradually propagates across the network. Once validated using network-wide rules, the transaction is stored in a pool with other unconfirmed transactions.

Approximately once every 10 minutes (for Bitcoin), a set of transactions from the pool is included in a new block and is added to the ledger, then the block is sent to all other nodes in the network so they can also update. Nodes are incentivized to do this with newly minted cryptocurrency and compete with each other to publish the next block by 'mining', or trying to quickly solve a mathematical puzzle. This puzzle takes lots of computational power to solve and is often a challenge to find a hash function input with a specific output[15], searching for special prime number chains[11]. If a miner node attempts to publish a new block without proof that it has solved the puzzle, the block will be rejected by the network. Publishing a new block is also known as mining the block.

Proof of Work's strength is its ability to attain consensus with thousands of untrusted notes in a completely permissionless environment. To attack a PoW blockchain such as Bitcoin, the adversary must control at least 51% of the network which is a nearly impossible task. However, with smaller networks such as Ethereum Classic, this attack becomes more possible for entities that possess a large amount of computing infrastructure[3].

Moreover, PoW has many disadvantages which severely hinder scalability. [8] accurately predicted that Bitcoin would require an enormous amount of electricity to run due to thousands of miner nodes competing to publish the next block. The network now consumes more energy than many countries, such as Romania. It takes about 10 minutes for a transaction to be added to the network, and only the transactions that pay the highest fees to the miner nodes are selected from the pool. This results in a maximum of 7 TPS compared to Visa's 4000 TPS, and these transactions are limited to expensive transfers because of high fees. Finally, even after a transaction has been successfully added to the network, a fork in history could occur, which happens when multiple miners solve the PoW challenge at the same time and each half of the network accepts a different block. To avoid this, many transaction handlers require 6 blocks to be added to the network before accepting the transaction, giving a transaction confirmation latency of about one hour. The combination of these weaknesses limits Proof of Work blockchains such as Bitcoin from having high throughput or usability.

*4.1.2 Bitcoin-NG.* When trying to increase Bitcoin's transaction throughput, a common suggestion by the community is to increase transaction block size. A 4 MB block will hold 4 times as many transactions as the default 1 MB; however, larger block size increases latency between nodes as it is propagated throughout the network and may require high-powered hardware. This on its own can actually reduce throughput and has not been adopted by Bitcoin after many proposals.

**Bitcoin-NG**[7] (Next-Generation) is a blockchain that is compatible with Bitcoin but modifies its PoW consensus protocol to overcome the scalability limits that Bitcoin faces. Rather than changing the block size or decreasing block interval, Bitcoin-NG changes the consensus protocol to be forward-looking: approximately every 10 minutes, a leader is selected to validate and publish transactions as soon as they are received. This eliminates the concern of block size and allows for unlimited throughput, as transactions are instead published in many micro-blocks every interval. Leaders are still selected through the same PoW computationally-intensive puzzles, and a special block called the key-block marks the change from one leader to another. While this improves transaction scalability and brings confirmation latency to a few seconds, it remains energy inefficient.

*4.1.3 Proof of Stake.* **Proof of Stake (PoS)**[17] is an early replacement meant to fix some of the problems with PoW. It also allows thousands arbitrary servers to run node software and is used by many blockchains such as Ethereum[20] and Ouroborus[10]. However, instead of using computationally expensive puzzles to choose who publishes each block of transactions, authority is distributed based on how much cryptocurrency each user owns. Miners 'stake' value in the system in order to confirm a percentage of all transactions and earn miner rewards. If a miner misbehaves, the network can take or void its cryptocurrency stake and prevent the node from mining another block, which encourages nodes to stay honest and increases security.

Due to this alternate block confirmation strategy, PoS is much more scalable than PoW in terms of energy efficiency, as nodes do not need to compete using computational force. This also lowers transaction confirmation latency as transactions can be added more quickly to the blockchain. Unlike PoW, in which a user could amass enough hardware to control 51% of the network to take it over, a PoS attacker would have to buy 51% of all available cryptocurrency to control a majority of the network, and at that point their stake would disincentivize them from harming the network.

However, some argue that PoS reduces decentralization since rich organizations with the most stake can make technical changes without consulting the community. Furthermore, while confirmation latency is improved, transaction throughput typically remains low due to the high number of nodes that receive and validate each new block.

*4.1.4 Delegated Proof of Stake.* **Delegated Proof of Stake (DPoS)** is a modified PoS algorithm developed for the BitShares[4] blockchain but is now used by many other blockchains as well. DPoS combines stake-based voting power with delegate elections so that shareholders elect which nodes (known as delegates or witnesses) will

[3] http://www.coinfox.info/news/reviews/6417-proof-of-work-vs-proof-of-stake-merits-and-disadvantages

[4] https://bitshares.org/

create blocks. Delegates usually take turns creating blocks every few seconds, and unresponsive or malicious delegates will quickly be removed from the cycle and have their altcoin stake confiscated, providing a high level of security.

DPoS retains all of the scalability advantages of PoS but requires less financial input from ordinary users, lowering the barrier of entry and making the system more democratic in nature. There is always an odd number of delegates in order to decrease the probability of long forks[12] and short block time keeps confirmation latency low. This algorithm provides faster processing of transactions that PoS or PoW which makes it a strong candidate for scalable public blockchains. However, the same problem still exists in which the rich may be able to control the blockchain to their advantage. The hope here is that the majority of shareholders will vote for delegates who give them the most rewards and act fairly to prevent this.

*4.1.5 Proof of Elapsed Time (PoET).* **Proof of Elapsed Time** is an algorithm proposed by Intel which uses trusted computing to enforce random waiting times before creating each new block. Trusted computing uses special hardware properties to protect critical code execution from being tampered with by an adversary, and is offered by major vendors such as Intel's SGX. Each node generates a random number to determine waiting time before it can create a new block. As part of publishing a block, the trusted hardware helps generate a proof of the waiting time[6].

PoET's two main advantages are efficiency and fairness. It is efficient in that it avoids all computational burdens imposed by Proof of Work and fair in that every server gets one vote, unlike Proof of Stake. However, a security anaylsis shows that Intel SGX may be vulnerable to security attacks, and that an adversary can potentially hijack the system by simulating the fastest honest node, even if very few nodes are compromised[6].

## 4.2 Byzantine Fault Tolerant Algorithms

Byzantine Fault Tolerant (BFT) protocols are a group of protocols that attain high resilience of node misbehavior, whether unintentional or malicious[19]. Unlike previous protocols, nodes in BFT protocols can independently identify and ignore any malicious or incorrect messages sent from other nodes often to the point of 1/3 of the network misbehaving. These algorithms frequently require a known set (or quorum) of nodes to operate, thereby suitable for private and consortium blockchains but incompatible with public blockchains. This impressive resilience is made possible by nodes sending synchronous gossip messages back and forth for all new transactions, status updates for nodes, new nodes, and several rounds of messages each time a new block is accepted.

*4.2.1 Practical Byzantine Fault Tolerance.* **Practical Byzantine Fault Tolerance (PBFT)**[5] is an early, popular algorithm that uses a 3-stage block commit process in which two thirds of the nodes must accept the block at each stage for the block to be accepted. In this algorithm, a leader is selected to initiate each commit process and this leader may rotate over time. The leader does not typically receive a reward. Tendermint is one blockchain that uses PBFT as its consensus algorithm.

*4.2.2 Redundant Byzantine Fault Tolerance.* **Redudant Byzantine Fault Tolerance (RBFT)**[1] add further security guarantees to PBFT by running $n$ leader-follower copies of the algorithm at once, with one primary copy that actually appends to the blockchain and the other copies used for increased resilience. Using Plenum, the blockchain can sustain up to 1/3 of malicious network nodes while experiencing only a 3% reduction in performance. Another algorithm that uses of slightly modified version of RBFT is Hyperledger Indy Plenum[5].

*4.2.3 Discussion.* BFT algorithms excel in their high availability and resiliency to significant attacks. However, heavy communication between nodes results in decreased performance as the number of nodes increases. PBFT and Plenum block confirmation have a time complexity (or latency) of $O(n^2)$ and $O(n^3)$ with respect to the number of nodes in the quorum, limiting the maximum number of nodes with high transaction throughput to the low-to-mid teens. Satisfying this, however, and with sufficiently powerful hardware, transaction throughput can theoretically be reasonably high and confirmation latency extremely low.

*4.2.4 Ripple.* Another algorithm related to BFT is the **Ripple Protocol Consensus Algorithm (RPCA)**[18] which lowers network latency by using completely asynchronous messaging between nodes. Some mistakenly claim that asynchronous protocols are impossible, overstating the FLP Impossibility Result[9], but in reality, this paper proves only that asynchronous *deterministic*, fault-tolerant protocols are impossible, and this protocol is technically nondeterministic but effective. It also minimizes messages between nodes overall which further improves both transaction throughput and all forms of latency. This allows ledgers using Ripple to run a slightly higher number of known nodes than other BFT algorithms—there are 34 validator nodes on the Ripple network at the time of writing and Ripple claims it can handle "150+" nodes[6]. Ripple makes the trade-off of increased transaction scalability for lower security and resilience, as the protocol only guarantees correct operation of less than 1/5 of the of the network is malicious. Another asynchronous protocol with similar properties is HoneybadgerBFT[14].

*4.2.5 Stellar.* The **Stellar Consensus Protocol**[13] uses Federated Byzantine Agreement (FBA) as a new approach to consensus. For FBA, each node commits a new transaction only if a large majority of nodes that it deems trustworthy agrees that the transaction should be committed. Stellar started as a fork of Ripple but is now fundamentally different as it allows anyone to participate in consensus with their own node, similar to PoW. It also does not provide any incentive scheme for running a node or good behavior.

Stellar has low latency and could potentially support thousands of transactions per second and a lab test by Deloitte achieved 10,000 TPS[7], though this TPS is unlikely for real-world systems. It can be argued whether or not Stellar increases or decreases security and immutability of transactions since each node can choose which other nodes it trusts. While this lowers the chance of interactions

[5] https://github.com/hyperledger/indy-plenum/wiki

[6] April 6th, 2020, pulled from Ripple's website: https://ripple.com/xrp/

[7] https://www.americanbanker.com/news/how-barclays-aims-to-bring-a-billion-unbanked-into-the-fold

with newly added malicious nodes, the network also becomes more centralized and could theoretically support multiple forks of history.

*4.2.6 Casanova.* **Casanova**[4] is a leaderless consensus protocol designed for permissioned blockchains (blockchains that can only be accessed if given permission). Casanova is often called an optimistic protocol because it takes advantage of the fact that most transactions are not double spend transactions. Whereas most protocols come to consensus after every transaction, Casanova uses a conflict exclusion protocol to run a choice consensus algorithm only when a double-spend is detected and only for conflicting transactions. Its paper also observes that transactions are usually unrelated to other transactions published at the same, and thus a partial ordering of transactions will suffice. It does this by linking blocks with a *directed acyclic graph (DAG)* instead of a chain. Because blocks need not be in a sequential chain, this allows the blockchain to process many transactions and blocks at once[3].

Casanova has the potential to support a very high TPS in an efficient manner; however, it has not been implemented and tested for either performance or security at the time of writing. If it performs according to the specification, it and other DAG-based consensus algorithms may become the new standard for blockchains. Casanova also tolerates a certain number of Byzantine nodes which can be adjusted by the ratio of validator nodes to total nodes in the network.

## 5 CONCLUSION

One of the biggest obstacles in the way of Blockchain's widespread adoption is scalability. Much research has been done to address this for both public blockchains and consortium/private blockchains. Public blockchains typically use a Proof-of-X algorithm in which nodes are not trusted and which benefits from high decentralization of authority. However, the large network size may limit the maximum transaction throughput of the network and add unacceptable latency which prompts some groups to use consortium blockchains instead. These blockchains can take advantage of permissioned BFT consensus algorithms which can provide increased security and performance, but with a maximum number of nodes and lower decentralization as a tradeoff. Some protocols such as the Stellar Consensus Protocol and Casanova that break these norms, but not all of these have yet been thoroughly tested.

Possible areas of future work include lowering the time complexity of BFT algorithms while maintaining their fault tolerance capability. This would allow more nodes to participate in consensus, thus further decentralizing authority. Another focus is continued work on permissionless BFT protocols, which would open up the security and resilience advantages to public blockchains.

Based on increased fault tolerance, efficiency and transaction scalability of BFT algorithms, we recommend that interested parties that can use consortium blockchains do so with one of these algorithms. However, for communities that require complete, public decentralization of trust, there are still several consensus protocols available that significantly improve on Nakamato's original Proof of Work algorithm.

## REFERENCES

[1] Pierre-Louis Aublin, Sonia Ben Mokhtar, and Vivien Quéma. Rbft: Redundant byzantine fault tolerance. In *2013 IEEE 33rd International Conference on Distributed Computing Systems*, pages 297–306. IEEE, 2013.

[2] Adam Bates, Dave (Jing) Tian, Kevin R.B. Butler, and Thomas Moyer. Trustworthy whole-system provenance for the linux kernel. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 319–334, Washington, D.C., August 2015. USENIX Association.

[3] Kyle Butt and Derek Sorensen. Streamlining classical consensus.

[4] Kyle Butt, Derek Sorensen, and Michael Stay. Casanova. *arXiv preprint arXiv:1812.02232*, 2018.

[5] Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance. In *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, OSDI '99, page 173–186, USA, 1999. USENIX Association.

[6] Lin Chen, Lei Xu, Nolan Shah, Zhimin Gao, Yang Lu, and Weidong Shi. On security analysis of proof-of-elapsed-time (poet). In *International Symposium on Stabilization, Safety, and Security of Distributed Systems*, pages 282–297. Springer, 2017.

[7] Ittay Eyal, Adem Efe Gencer, Emin Gün Sirer, and Robbert Van Renesse. Bitcoin-ng: A scalable blockchain protocol. In *13th USENIX symposium on networked systems design and implementation NSDI 16)*, pages 45–59, 2016.

[8] P. Fairley. Blockchain world - feeding the blockchain beast if bitcoin ever does go mainstream, the electricity needed to sustain it will be enormous. *IEEE Spectrum*, 54(10):36–59, 2017.

[9] Michael J Fischer, Nancy A Lynch, and Michael S Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, 32(2):374–382, 1985.

[10] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017*, pages 357–388, Cham, 2017. Springer International Publishing.

[11] S. King. Primecoin: Cryptocurrency with prime number proof-of-work. Technical report, 2013.

[12] Daniel Larimer. Dpos consensus algorithm - the missing white paper. https://steemit.com/dpos/@dantheman/dpos-consensus-algorithm-this-missing-white-paper, 2017.

[13] David Mazieres. The stellar consensus protocol: A federated model for internet-level consensus. *Stellar Development Foundation*, 32, 2015.

[14] Andrew Miller, Yu Xia, Kyle Croman, Elaine Shi, and Dawn Song. The honey badger of bft protocols. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 31–42, 2016.

[15] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Technical report, 2008.

[16] Scott Ruoti, Ben Kaiser, Arkady Yerukhimovich, Jeremy Clark, and Robert Cunningham. Sok: Blockchain technology and its potential use cases, 2019.

[17] Fahad Saleh. Blockchain without waste: Proof-of-stake. *Available at SSRN 3183935*, 2020.

[18] David Schwartz, Noah Youngs, and Arthur Britto. The ripple protocol consensus algorithm. *Ripple Labs Inc White Paper*, 5(8), 2014.

[19] Marko Vukolić. The byzantine empire in the intercloud. *SIGACT News*, 41(3):105–111, September 2010.

[20] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32, 2014.