Analysis of Proof of Stake flavors with regards to The Scalability Trilemma

Paul Schaaf, Filip Rezabek*, Holger Kinkelin*
*Chair of Network Architectures and Services, Department of Informatics
Technical University of Munich, Germany
Email: ge75sab@mytum.de, frezabek@net.in.tum.de, kinkelin@net.in.tum.de

***Abstract*—Blockchains are a rapidly evolving area of research and experimentation. Since Bitcoin's introduction in 2008, different protocols have been proposed and implemented with the goal of improving on Bitcoin's core feature, the Proof of Work consensus mechanism. A critical area many of the newer mechanisms focus on is a reduction in energy usage, for example.**

This paper presents and compares different Proof of Stake (PoS) mechanisms – an increasingly popular alternative to Proof of Work – that have been developed in recent years. The focus of our comparison are the mechanisms' abilities to solve the Scalability Trilemma, that is, a consensus mechanism's ability to achieve decentralization, security, and scalability. We find that Unbonded PoS is the most decentralized mechanism but comes with vague security assumptions. Bonded PoS is more secure at the cost of decentralization. Lastly, Delegated PoS achieves scalability but suffers from low decentralization and security.

***Index Terms*—blockchains, proof of stake, scalability trilemma**

1. Introduction

Any given blockchain is a complex distributed system and the result of a variety of design decisions. One of the most important considerations is a blockchain's consensus mechanism. In decentralized networks where each node stores all the state (because there is no central server), there is a need for a mechanism that allows all nodes to come to consensus on which state changes should be applied to the current state and in which order they should be applied. This is so that after a state change all nodes have saved the same state [1].

In 2008, Satoshi Nakamoto used the concept of Proof of Work (PoW) to create Bitcoin's consensus mechanism. This mechanism makes nodes compete to solve hash puzzles (which is called *mining*) for Bitcoin rewards and the right to propose a specific set of changes [2].

Since then, Bitcoin's consensus mechanism has been criticized, mainly for its large consumption of energy because nodes are incentivized to buy more machines as long as the rewards outweigh the energy costs [3].

In an attempt to improve on the PoW consensus mechanism, other mechanisms have been researched and implemented in other blockchains. Most notably, Proof of Stake (PoS), a mechanism that removes the advantage of owning more machines and instead directly uses capital. Instead of mining, it employs a randomness function to choose the next block proposer which favors those with more capital. While PoS does not increase the fairness of the rewards distribution, it does come with lower energy requirements [4].

It is the goal of this paper to provide answers to the questions of what the subcategories of PoS are and how they differ. In sections 2 and 3 respectively, we present relevant background knowledge and highlight related work. Equipped with this knowledge, we compare the different flavors in section 4. In particular, we compare Unbonded Proof of Stake (UPoS), Bonded Proof of Stake (BPoS), and Delegated Proof of Stake (DPoS). The focus of this comparison is the ability of each mechanism to solve the so-called *Scalability Trilemma*. The trilemma states that it is difficult for any mechanism to achieve scalability without sacrificing security or decentralization [5]. It is these three properties by which we compare the mechanisms. In section 5, we summarize our findings and point to potential areas for future work.

2. Background

This section provides a high-level overview of how a blockchain works, presents the Proof of Work and Proof of Stake consensus mechanisms, and introduces the Scalability Trilemma.

2.1. Structure and Participants in a Blockchain System

The two main parties in a blockchain system are the users – those that wish to send transactions – and the nodes – those that maintain the blockchain. The blockchain is a log that contains all the transactions that have been committed so far. The transactions are aggregated in so-called *blocks*. Each block has a reference to the previous block, forming a chain of blocks (thus forming a special form of a linked list). The blockchain is kept as a local copy by every node because there is no central server. The current state of the network can be computed by replaying all transactions in the blockchain. Thus, every transaction moves the blockchain from one state into a different one. Using the example of cryptocurrencies, this state might save users' money which can be sent to other users by submitting transactions [6, Chapter 1-2].

2.2. Consensus

In Section 2.1 we have established that all nodes need to save an identical copy of the blockchain. Thus,

doi: 10.2313/NET-2022-01-1_13

when users request to send transactions, all nodes need to agree on which of the requested transactions should be added to the chain next and in which order, that is, what the next block to be added to the chain should look like. Allowing nodes to reach consensus is the goal of a consensus mechanism. Blockchain consensus mechanisms need to function in adversarial environments, that is, those in which there are malicious nodes [6, Chapter 2]. In particular, these mechanisms should be *sybil-resistant*, that is, they must continue to work under the assumption that malicious actors can create nodes at no cost [7]. Hence, a simple direct democracy where each node represents one vote with the winner being elected to propose the next block does not work [6, Chapter 2].

Subsections 2.2.1 and 2.2.2 explore the PoW and PoS mechanisms which are sybil-resistant.

2.2.1. Proof of Work. Proof of Work was first introduced in 1993 to prevent service abuses in computer networks [8]. It allows someone to prove they have done some work with the verification of this proof being cheaper than the work itself. We use Bitcoin as an example to illustrate how PoW blockchains incorporate this concept. All nodes in the system hold an individual block of requested transactions that could be added to the chain next. Each node tries to find a number – the *nonce* – so that when added to its block, the block hashes to a value smaller than a value X set by the protocol. There is no efficient algorithm to do this, so a node can only try different numbers (this is the *Work* in Proof of Work). If a node finds such a number, it broadcasts the block. Each node that receives it verifies the PoW (which equates to confirming the block's hash is smaller than X, which can be done in constant time) and the transactions in the block (e.g. the sender of some money must have agreed to the transfer). If valid, a receiving node adds the block to its local chain and starts working on the next one. This process is called *mining*. The result of this mechanism is that in regular intervals a new node is chosen to propose the new block. Note that a node is only chosen implicitly by finding the nonce; there is no voting taking place.

A benefit of PoW is that adding additional nodes does not increase the likelihood of finding the nonce, only the computing power matters. Hence, the Proof of Work consensus mechanism is sybil-resistant. In addition, it follows that the more distributed the computer power is across nodes, the more decentralized - and thereby more resistant to attacks - the network is [6, Chapter 2-4].

2.2.2. Proof of Stake. Proof of Stake is designed to be an alternative for PoW and was first proposed in a forum post in 2011 [9]. It works by essentially simulating the mining process. While Bitcoin's Proof of Work was described as *"essentially one-CPU-one-vote"* in the original whitepaper [2] by Nakamoto, Proof of Stake maps one unit of currency to one vote. *Validators*, as they are commonly called in PoS chains, deposit the respective blockchain's currency (in blockchain jargon: they *stake* and their capital is *staked*) and one validator is then chosen by a function to propose the next block. While incorporating some form of randomness, this function is more likely to choose a validator with a higher stake (again, analogous to owning more machines in PoW). One advantage of PoS over PoW is that the amount of machines a node controls does not increase the likelihood of being chosen as the next block producer, leading to a drastically lower energy footprint (about two thousand times more energy efficient in some cases [10]) [11].

The Nothing at Stake Problem. PoS consensus mechanisms have to deal with one issue PoW mechanisms do not: the Nothing at Stake Problem.

In general, blockchains can fork, meaning that two valid blocks are proposed at the same time, turning the blockchain into a tree with the two leaves referencing the previous block. In Proof of Work, a node chooses one branch by devoting its computing power to finding the nonce for the next block that references that branch's leaf. It could also use 30% on one branch and 70% on the other. The main point is that a node cannot use more than 100% of its computing power. A fork in PoW does not alter a miner's ability to produce blocks because the resource securing the network (the computers) is outside the network. One branch will likely have more computing power supporting it and eventually all nodes switch to that branch [6, Page 209].

The story is different in Proof of Stake, however [12]. A validator's likelihood to be a block producer is influenced by their stake, that is, the amount of the blockchain's native currency that they have staked. This means that the resource securing the network is part of the blockchain itself. Hence, when a fork happens, this resource is duplicated. Each validator wants to participate in consensus on the branch of the fork that eventually becomes the main branch because the rewards for participating on the eventually abandoned branch will not be considered real. However, because their capital is duplicated, they do not have to choose like block producers in PoW do, they can just produce blocks on both branches. It is trivial to see that if every validator thinks like this, there will never be a main branch because all branches continue producing blocks. After all, from the viewpoint of a single validator, there is no penalty for acting like this, a penalty that would e.g. remove the validator's funds. As a result, the validator has *nothing at stake* that they could lose if they act like described above. The reason this is a problem is that if forks do not clear up after some time, no consensus is reached and users of the currency cannot be sure their transactions are final.

2.3. The Scalability Trilemma

The Scalability Trilemma states that it is incredibly difficult for a blockchain to be scalable while staying secure and decentralized [5]. Note that for none of these properties there exists a single metric and a single value for that metric to achieve the property. This means in some cases it might be difficult to determine if one chain is, for example, truly more decentralized than the other. However, when comparing approaches where the differences are big enough, employing the trilemma makes sense. With this in mind, it is worth briefly exploring the three properties of the trilemma and which metrics could be used to measure them.

2.3.1. Decentralization. The rationale for decentralization is that in a peer-to-peer protocol, allowing one actor

to take sole control is likely not in the interest of the other nodes. Two values that are often looked at are the number of nodes in a system and how many of them are controlled by a single entity [5].

2.3.2. Security. This property is the most obvious one. Any serious approach towards establishing a cryptocurrency should come with safety guarantees. A network should result in (possibly probabilistically) final transactions and be resistant to attacks.

2.3.3. Scalability. The two metrics that are most important when describing a blockchain's ability to scale are throughput and confirmation latency. That is, how many transactions can be executed per second and how long do users have to wait until they can consider a transaction non-reversible [3].

3. Related Work

With Bitcoin having been invented only in 2008, 13 years ago, research specifically on blockchains is still in the early stages (although many ideas Bitcoin and other blockchains rely on come from well-researched distributed systems theory and cryptography).

In addition, in this industry, much of the research that has been done has not necessarily been published in academic journals but in blog posts or projects' whitepapers. A relevant example: Vitalik Buterin – co-founder of Ethereum, the 2nd biggest cryptocurrency by market cap – reasons for Proof of Stake and how it relates to Proof of Work, and presents other related concepts like the Nothing at Stake Problem on his blog [13].

With that said, some more formal research has also been carried out. Narayanan et al. provide a thorough introduction to Bitcoin and cryptocurrencies [6]. Lepore et al. compare PoW, PoS and Pure Proof of Stake^{TM} and provide a framework for future comparisons [14]. Nguyen et al. present differences between the consensus algorithms of specific protocols (as opposed to the broader approach we take in this paper) and analyse staking pools, a phenomenon on PoS blockchains similar to mining pools on the Bitcoin network [4].

4. Recent Advances in PoS

This section focuses on recent advances in PoS algorithms. It first considers different approaches to solving the Nothing at Stake problem. Then, it explores Delegated PoS, a mechanism to increase the scalability of Proof of Stake. It also shows how each of these adjustments moves a protocol along the three dimensions of the Scalability Trilemma.

4.1. Solving the Nothing at Stake Problem

This section highlights two mechanisms that aim to solve the Nothing at Stake Problem explained in section 2.2.2 and describes the effects both mechanisms have on the properties of the Scalability Trilemma.

4.1.1. Bonded Proof of Stake. One solution to the Nothing at Stake Problem is *Slashing* [15]. The idea behind it is that nodes should only be allowed to produce a block if they have something to lose. To this end, the protocol requires a node to agree that if someone can prove that the node produced a block on two different forks (or more generally, act maliciously), the node loses part of its staked capital.

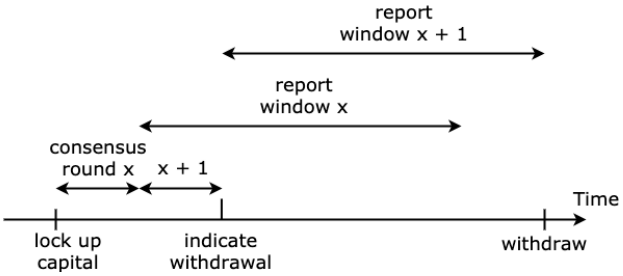

Figure 1: Example BPoS Participation Flow

Figure 2 illustrates this process, using an imaginary BPoS protocol. First, the participant locks up capital and begins participating in several consensus rounds - two rounds in this example (an unrealistically low number only chosen to conserve space). As shown in the figure, there is a report window for each consensus round in which others may report the node for malicious behaviour. This means that after the staker indicates their wish to withdraw, there is a period in which they have to wait and do not earn any rewards. Hence, their capital is *bonded*. Only once this period has passed, may they withdraw.

BPoS and the Scalability Trilemma. Bonded Proof of Stake is an addition to regular PoS that is meant to fend off the Nothing At Stake Problem. Hence, BPoS increases a blockchain's security.

However, BPoS introduces factors that might lead to centralization. First, the requirement to lock up capital limits the set of nodes to people that can afford to lock up their disposable income for such a purpose. Secondly, some PoS protocols allow users to delegate native currency (not to be confused with DPoS) to validators if they cannot or do not want to run their own node. This way, users can receive some of the maintenance rewards, while the validators take a fee. Generally, it is advised to split the delegations across multiple validators to maintain the decentralization of the network. In networks that employ slashing, however, users will think twice about whom they are delegating their money to. In practice, this has led to stake being concentrated across a few trusted companies in many PoS protocols [16], [17].

Bonded Proof of Stake is only meant to make a blockchain more resistant and has, in itself, no effects on scalability.

BPoS in Practice. The most prominent protocol using Bonded Proof of Stake is Ethereum 2.0 (ETH2) [18]. To become a validator, a deposit of 32 ETH (worth about $55,000 as of June 13, 2021 [19]) is required. If a validator produces a block on two forks, it is slashed whereas the amount slashed increases the more validators act maliciously. If a third of validators act maliciously at roughly the same time, their entire deposit is slashed [20], [21].

doi: 10.2313/NET-2022-01-1_13

4.1.2. Unbonded Proof of Stake. Another way to approach the Nothing at Stake problem is to bypass it by using different assumptions. The most important one is that the average member of society is virtuous; in particular, they do not wish to hurt society or its monetary system. There may be parts of society that act maliciously but that part is small in any functioning society. Thus, assuming that the node distribution in a blockchain resembles society, most nodes will not produce blocks on two different forks or be bribed to do so because this would hurt the blockchain's health. Hence, there is no need to lock up capital and punish bad behavior because it will never happen on a scale large enough to affect the network anyway [22], [23].

UPoS and the Scalability Trilemma. UPoS moves the protocol using it further towards decentralization because there is no lockup of capital. However, it relies on more assumptions than Bonded Proof of Stake, mainly that the average member of a functional society is virtuous and that a blockchain's node distribution resembles society. Especially the latter assumption is debatable because in a public UPoS blockchain system, nodes are anonymous and there are no punishments, unlike most human societies. Open questions like these make it impossible to give a clear answer on the mechanism's security for now.

Analogous to BPoS, UPoS, in itself, does not affect scalability.

UPoS in Practice. Algorand is a blockchain using a mechanism called Pure Proof of Stake™ (PPoS) [22]–[24]. PPoS does not require any lockup of capital. However, available capital is still used to grant proportional voting power to nodes in the network. Each round, a chosen node's block is voted on by random committees of nodes. Using a subset of all nodes to add the next block increases efficiency. The random composition of these committees makes it highly likely that they resemble the overall network. To find a node's role for the creation of the next block (e.g. block proposer or committee member), it executes a local verifiable random function that requires no communication with other nodes. This can result in multiple nodes proposing a block. However, only the block with the highest priority (which it is more likely to have the more of the native currency the proposers owns) will be accepted by the committee. This results in nodes only having a single block to vote on which they do if it is valid and they are honest.

4.2. Delegated Proof of Stake

Delegated Proof of Stake is a consensus mechanism that combines PoS with a governance system [25]. Its goal is to increase a PoS blockchain's scalability. Its main innovation is that it only allows a fixed number of representative nodes to participate in consensus. With only a small number of nodes required to come to consensus, the communication overhead drops and consensus is reached faster.

DPoS consists of two steps. First, an election takes place to determine the representative nodes. All nodes can participate and they use the blockchain's native currency to vote. From the election onwards, the blockchain may use another PoS mechanism to let the elected nodes come to consensus for several blocks after which a new election takes place. Hence, this mechanism has similarities to a representative democracy.

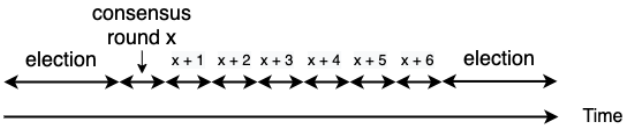

Figure 2: Example Delegated Proof of Stake

Figure 2 shows an imaginary DPoS protocol where an election happens every 7 rounds. Important to note, the election round takes longer than the consensus rounds because it has more participants.

DPoS and the Scalability Trilemma. DPoS provides a fixed and low number of consensus nodes, thereby increasing a blockchain's ability to scale. However, this benefit comes at the cost of decentralization and security. It is trivial that any system in which a small, fixed number of nodes have all the control is less decentralized than a system in which control is spread across an unlimited number of nodes. In addition, even if the elected nodes are not malicious, attacking a low, fixed number of nodes is easier than attacking a large number of nodes that are possibly not known in advance.

DPoS in practice. This paragraph briefly showcases the best known DPoS blockchain: EOS [26] (this is indeed its name, not an abbreviation). In EOS, a new block is produced by one of 21 consensus nodes every 0.5 seconds. Next to the 21 consensus nodes that produce blocks, there are about 530 other nodes on standby that could be elected as of June 15, 2021. The consensus nodes receive EOS (the EOS blockchain's native currency) as a reward for maintaining the system. The approximately 40 nodes on standby with the highest number of votes also receive rewards, the others do not. A new election for the consensus nodes happens every 126 blocks. Consensus nodes use the asynchronous Byzantine Fault Tolerance (aBFT) consensus algorithm which ensures transaction finality after 1 second. The maximum observed transactions per second to date on EOS is about 4000 [27]. This example demonstrates that EOS is indeed more scalable than, for instance, Bitcoin and Ethereum as of June 15, 2021 (with about 7 and 15 tps, respectively [28]). However, EOS has received some criticism regarding malicious behavior by its consensus nodes. In particular, allegations of vote buying (a consensus node bribes token holders to vote for it) and mutual voting (consensus nodes with large amounts of EOS agree to vote for each other to stay in power) have been made public [29], [30].

5. Conclusion and Future Work

In this paper, we have presented and compared different flavors of the Proof of Stake consensus mechanism with regards to their ability to solve the Scalability Trilemma.

In sections 1 and 2 we motivated the need for consensus mechanisms, which allow nodes in a distributed system to agree on state changes. We then presented two broad categories of consensus mechanisms, the Proof of Work mechanism – employing computing power to

doi: 10.2313/NET-2022-01-1_13

reach consensus – and the Proof of Stake mechanism – employing capital to reach consensus, thereby achieving a lower energy footprint.

In section 4 we compared flavors of the Proof of Stake consensus mechanism. The metrics of the comparison were the three properties of the Scalability Trilemma. The summarized findings can be found in table 1.

Mechanism	Scalability	Decentralization	Security
BPoS	0	-	+
UPoS	0	0	0
DPoS	+	-	-

TABLE 1: Comparison of PoS flavors, 0 ≙ base case, + ≙ increase, - ≙ decrease

UPoS is used as the base case from which the other two are compared in the table because as we have shown it is the mechanism which is the easiest to achieve – it only requires a change of assumptions. We have found that BPoS alone does not offer any scalability improvements over UPoS whereas DPoS does, at the cost of both decentralization and security. Punishable capital requirements in BPoS increase security at the cost of decentralization. An important takeaway from this comparison is that none of the presented flavors can satisfy all three properties of the Scalability Trilemma.

At this point, it is important to mention, however, that these three flavors represent three sets of mechanisms with each set containing blockchains that still differ significantly. In addition, while the two sets of BPoS and UPoS mechanisms are mutually exclusive, BPoS/UPoS and DPoS are not. Hence,one should, for example, not conclude that a blockchain using UPoS cannot scale. A mechanism that is only in the UPoS set may still scale by employing other technologies but this increase in scalability is not caused by UPoS itself. One such example is Algorand, discussed in section 4.1.2, which increases scalability through short-lived stake-weighted random committees (as opposed to elected longer-lived ones in DPoS).

As a result, comparing these broad categories in a vacuum is only the tip of the iceberg. A more practical comparison in the future might compare how these mechanisms are used in conjunction with other technologies in different protocols. In addition, a more holistic analysis would also include innovations that are above the protocol level. In BPoS, for example, there exists the concept of Liquid Staking on the application layer. It is meant to alleviate the problems with locked-up capital by allowing stakers to withdraw staked capital immediately for a fee [31].

References

[1] "A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks," *IEEE Access*, vol. 7, pp. 22 328–22 370.

[2] S. Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. Accessed: 15/05/2021. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[3] Q. Zhou, H. Huang, Z. Zheng, and J. Bian, "Solutions to Scalability of Blockchain: A Survey," *IEEE Access*, vol. 8, pp. 16 440 – 16 455, Jan. 2020.

[4] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, D. Niyato, H. T. Nguyen, and E. Dutkiewicz, "Proof-of-Stake Consensus Mechanisms for Future Blockchain Networks: Fundamentals, Applications and Opportunities," *IEEE Access*, vol. 7, pp. 85 727–85 745, 2019.

[5] A. Altarawneh, T. Herschberg, S. Medury, F. Kandah, and A. Skjellum, "Buterin's Scalability Trilemma viewed through a State-change-based Classification for Common Consensus Algorithms." IEEE, 2020.

[6] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, 2016.

[7] B. N. Levine, C. Shields, and N. Boris Margolin, "A Survey of Solutions to the Sybil Attack," 2005.

[8] C. Dwork and M. Naor, "Pricing via Processing or Combatting Junk Mail," E. F. Brickell, Ed. Springer-Verlag, 1992, p. 139–147.

[9] QuantumMechanic. Proof of stake instead of proof of work. Accessed: 15/05/2021. [Online]. Available: https://bitcointalk.org/index.php?topic=27787.0

[10] A country's worth of power, no more! [Online]. Available: https://blog.ethereum.org/2021/05/18/country-power-no-more/

[11] F. Saleh, "Blockchain Without Waste: Proof-of-Stake," *Review of Financial Studies*, vol. 34, pp. 1156–1190, 2021.

[12] V. Buterin. (2014) On Stake. Accessed: 15/05/2021. [Online]. Available: https://blog.ethereum.org/2014/07/05/stake/

[13] ——. Accessed: 06/06/2021. [Online]. Available: https://vitalik.ca/

[14] C. Lepore, M. Ceria, A. Visconti, U. P. Rao, K. A. Shah, and L. Zanolini, "A Survey on Blockchain Consensus with a Performance Comparison of PoW, PoS and Pure PoS," *mathematics*, vol. 8, 2020.

[15] V. Buterin. (2014) Slasher: A Punitive Proof-of-Stake Algorithm. Accessed: 15/05/2021. [Online]. Available: https://blog.ethereum.org/2014/01/15/slasher-a-punitive-proof-of-stake-algorithm/

[16] Cosmos Validators Leaderboard. Accessed: 13/06/2021. [Online]. Available: https://web.archive.org/web/20210613103151if_/https://cosmos.fish/leaderboard/all

[17] Eth1 Deposit Addresses. Accessed: 13/06/2021. [Online]. Available: https://web.archive.org/web/20210613103156/https://beaconcha.in/charts/deposits_distribution

[18] eth2. Accessed: 15/05/2021. [Online]. Available: https://ethereum.org/en/eth2/

[19] Ethereum USD Historical Data. [Online]. Available: https://www.coingecko.com/en/coins/ethereum/historical_data/usd#panel

[20] V. Buterin and V. Griffith, "Casper the Friendly Finality Gadget," 2017.

[21] Serenity design rationale. Accessed: 13/06/2021. [Online]. Available: https://web.archive.org/web/20210613112919/https://notes.ethereum.org/@vbuterin/rkhCgQteN#Slashing-and-anti-correlation-penalties

[22] Various questions about the algorand blockchain. Accessed: 05/06/2021. [Online]. Available: https://medium.com/algorand/various-questions-about-the-algorand-blockchain-ef8bf719f1f

[23] Silvio micali's lecture on algorand. Accessed: 05/06/2021. [Online]. Available: https://youtu.be/NykZ-ZSKkxM

[24] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, "Algorand: Scaling byzantine agreements for cryptocurrencies." Association for Computing Machinery, 2017, pp. 51–68.

[25] Z. Zheng, H.-N. Dai, and S. Xie, "Blockchain challenges and opportunities: A survey," *International Journal of Web and Grid Services*, vol. 14, Jan. 2018.

[26] EOS Consensus Protocol. Accessed: 15/05/2021. [Online]. Available: https://developers.eos.io/welcome/v2.0/protocol/consensus_protocol

[27] Bloks.io | fastest eos block explorer and wallet. Accessed: 15/05/2021. [Online]. Available: https://www.bloks.io/

[28] Sharding FAQs. [Online]. Available: https://eth.wiki/sharding/Sharding-FAQs

[29] Corrupt governance? what we know about recent eos scandal. Accessed: 15/05/2021. [Online]. Available: https://cointelegraph.com/news/corrupt-governance-what-we-know-about-recent-eos-scandal

[30] V. Buterin. Governance, part 2: Plutocracy is still bad. Accessed: 15/05/2021. [Online]. Available: https://vitalik.ca/general/2018/03/28/plutocracy.html

[31] M. Di Maggio. Liquid staking: A discussion of its risks and benefits. Accessed: 06/06/2021. [Online]. Available: https://web.archive.org/web/20210606192846/https://medium.com/terra-money/liquid-staking-a-discussion-of-its-risks-and-benefits-bbaa957d9233

doi: 10.2313/NET-2022-01-1_13