An Exploration of State-of-the-Art Blockchain Scalability Approaches

Kees Fani[1], Manuel Ferreira[2], and Cornel de Vroomen[3]

Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology

Email: {[1]K.Fani, [2]M.BorbadaSilvaFalcaoFerreira, [3]C.deVroomen}@student.tudelft.nl

***Abstract*—The limited throughput of many blockchain solutions prevent the adoption of this relatively new technology at larger scales. There have been many approaches to attempt to resolve this daunting issue. Some of the more structural and impactful manners of increasing scalability in blockchains are achieved by using a different kind of consensus algorithm. In this paper we shall compare these approaches by diving into a wide range of consensus algorithms and blockchain projects. Afterwards, we will discuss and compare these solutions and draw our own conclusions in terms of the scalability of the solutions.**

Keywords: blockchain, scalability, consensus, Byzantine Fault Tolerance, cryptocurrency

I. Introduction

The blockchain is a relatively new technology which has the potential to restructure our whole society [1]. The most promising part of a blockchain is that it is able to cultivate trust between two parties that do not know each other. This has the ability to remove the middle man that stands between a lot of transactions between two people done today. Many companies provide value by acting as the intermediary during transactions. Airbnb [2], Uber [3] and TaskRabbit [4], but also the more established middle men, such as the financial institutions or even the government are prominent examples.

Lately, there has been much interest in the blockchain and cryptocurrencies. With this greater interest, more and more users are using blockchain based platforms such as Steemit [5], a social media platform which monetarily incentives users to add quality content. However, the blockchain networks currently in place, usually cannot effectively handle the large stream of users entering the blockchain domain. Transactions often take longer and become more expensive. This is the case because Bitcoin's blockchain, set the precedence of upholding a relatively constant and limit in transactions per time period.

The biggest blockchain network by market cap, the Bitcoin network, had in its peak, transactions which on average would cost about 30 dollar [6]. Besides that, the time to get this transaction confirmed was at its peak about 3 hours [7]. The blockchains should become scalable, such that the fees stay affordable for every type of transaction, and that they get confirmed quickly.

This paper will review and compare current blockchain projects and scalability methods. The key question that will be answered is:
"**What are the current state-of-the-art methods to scale a blockchain?**".
This leads to the following sub-questions:

- What is fundamental to blockchain and why is the throughput bounded?
- What consensus protocols are there, and how scalable are these protocols?
- What other techniques are there to scale a blockchain?
- What techniques are the current large projects implementing to scale their blockchain, and how to they compare to each other?

II. Blockchain fundamentals and challenges

In order to understand blockchain scalability techniques, we need to establish a basic understanding of blockchain and its limitations. Here we will explain fundamental properties of blockchains, and we will explain the limitations and challenges current blockchains face.

A. Blockchain fundamentals

At the most fundamental level, a blockchain is a chain of blocks. Herein we distinguish two parts, the chain links and the blocks. The links of the blockchain connect the blocks one by one. This is done by having every block, include a hash of the block that came before it. The blocks themselves typically include a collection of transactions.

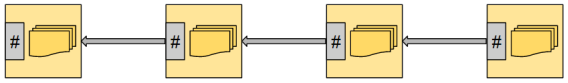

Figure 1: A simple blockchain

Chaining blocks up using hashes, ensures that no modifications can be done to the transactions, without having to change the respective block hashes. The reason blockchains use hashes to verify integrity, is that hashes are quick to create, which makes the hashed data easily verifiable. SHA-256 [8] is a hashing algorithm that is used often in blockchains.
A verification datastructure that is often used for transaction verification is the Merkle Tree. The Merkle tree allows for quicker and more targeted verifications of smaller parts of the data.
Valid blocks, thus transactions, can be appended to the blockchain. Every node has access to the blockchain. Every node can propose to add new blocks to a blockchain. These three attributes make blockchain a desirable solution for digital currencies and potentially other decentralized applications. [9]

B. Challenges

The protocols behind the blockchain networks have to make sure all copies of the blockchain are the same, they cannot be tampered with and keep the network as secure as possible. This guarantee is currently much harder to sustain without a middleman.

The current state-of-the-art blockchains are slow and behind in handling transactions, when compared to centralized systems. The Bitcoin blockchain for example, can only handle about seven transactions per second [10], similarly Ethereum, which normally handles 20 per second, now only handles around six transactions per second [11]. In 2013 Visa, which is an intermediary, handled on average 4000 credit card transactions per second, with a maximum of 47000 transactions per second. [12] This is not caused by a lack of computing power in decentralized systems. This gap in throughput capabilities between decentralized and centralized systems, is caused by the inherent structure of the network and the protocols behind it.

III. Blockchain consensus

Whenever important policies, regulations or values are defined, it important that all parties agree with their content. Appending upon these key aspects requires agreement from all parties involved.

This principal also applies to blockchain. In blockchain, the longest chain of valid blocks is generally regarded as the 'correct' chain. In other words, consensus is reached on the longest chain. All content of the blockchain should be valid and agreed upon. Nodes should never be able to modify or append blocks with malicious intent.

To uphold these principals, quite a few consensus algorithms have been developed. In this paper, we will cover some of the more prominently used and potentially effective consensus algorithms.

A. Byzantine Fault Tolerance

For distributed systems, it is essential to be Byzantine Fault Tolerant (BFT). BFT stems from a particular problem which can occur in these systems called the "Byzantine Generals Problem" [13]. The problem metaphorically describes computer systems as generals who want to conquer a city. To conquer the city, all generals need to agree to attack. To survive they all need to retreat. There could be traitors, who want to confuse the loyal generals. If more than one third of the generals are traitors, the attack of the city will not succeed.

Solving this problem is harder for fully decentralized public systems than centralized private ones. In a centralized private system there is one owner of all the computers in the network. The owner is the one who actually instructs the computers, and can therefore be fairly sure every computer complies to the rules.

In decentralized systems, you cannot trust the other nodes. Since you cannot trust the other generals in the Byzantine Generals Problem. Protocols need to be used such that a traitor cannot influence the outcome of the network. These consensus protocols make sure consensus over all nodes is reached, but dampen the throughput of transactions.

B. Practical Byzantine Fault Tolerance

The Practical Byzantine Fault Tolerance [14] (PBFT) consensus algorithm came out in 1999 and was a big breakthrough in distributed computing. Projects that make use of PBFT include Ripple and Stellar. In a distributed network we will get faulty results, when 1/3 of the nodes in the network are dishonest. For PBFT, we assume that there is a distributed network with $3f + 1$ nodes. Where f is the maximum amount of nodes that could be faulty. A client which wants to do an operation on the network, will send a request to all nodes in the network. If at least $f+1$ nodes give the same result, the result is assumed as the actual answer.

In a blockchain, consensus can be safely reached by using PBFT. There is a distinction between two types of nodes, a primary/validation node and normal nodes. Primary nodes are responsible for a group of nodes, and sends requests to this group when they receive a request. Under normal circumstances a transaction sent to the network, will end up in a particular primary node. Consensus will be reached when at least $f + 1$ nodes give the same result, in the group of the primary node. To achieve this, we compromise on anonymity of the nodes, as everyone needs to know the type of a node and to which group it belongs.

C. Proof of Work

The Proof of Work protocol, also called HashCash, is a consensus protocol used in Bitcoin and Ethereum. It was first introduced in 1992 with the paper 'Pricing via Processing or Combatting Junk Mail' by Dwork and Naor. Later in 1997 Adam Back proposed HashCash, a Proof of Work protocol to counter DDoS attacks. In short, the protocol imposes the user to first do some work, before appending a resource to the blockchain. The user has to create a proof of work, which is a solution to a 'puzzle' which takes work to solve, but is easy to verify.

Proof of Work as a consensus protocol in a blockchain network, was first implemented by Satoshi Nakamoto in Bitcoin [15]. In Bitcoin, miners have to solve a computational puzzle before they can submit a new block to the blockchain.

The difficulty of these puzzles is updated every 2016 blocks, based on a moving average of the amount of blocks per hour. The target amount is 6 blocks per hour. As you may notice, this does not sound very scalable. If this is the only protocol used, there would always be 6 blocks per hour. Given that the transactions and block size remain the same, there is no improvement possible in throughput.

D. Proof of Stake

While proof of stake is a secure way of reaching consensus in blockchains [16], it does have its shortcomings. There are two main issues with proof of work, that proof of stake seeks to address. The first issue is the massive amount of electricity needed [17] to solve the 'puzzles' necessary for proof of work. The second issue, is the fact that control over the system is not proportional to their stake of the blockchain's currency. Instead, a user's control of the blockchain is proportional to their share of computational power used to solve puzzles. This issue is also known as the 'Nothing at stake problem' [18].

The proof of stake consensus mechanism can be implemented in various ways. The Ourobous protocol [19] created by Kiayias, A. et al, implements proof of stake by building mathematical proofs, to randomly choose a single block producer. The Blackcoin [20] cryptocurrency uses a different mechanism to achieve proof of stake. This is done by using a combination of staked coins and the 'age' of these coins, to automatically solve a proof-of-work puzzle.

E. Delegated Proof of Stake

In many democracies around the world, people choose their representatives, to uphold their political beliefs and keep the country running effectively. Consensus on laws and policies is reached through votes and decisions between representatives.

Analogously, the same principles could be said about reaching consensus in the blockchain. In Delegated Proof of Stake, stake holders elect block producers to maintain the blockchain [21]. These elected block producers are often also called delegates. A user's voting power is proportional to their stake of coins in the respective blockchain.
An advantage of delegated proof of stake is its potential to be more effective and efficient than using a pure proof of stake protocol. [21] This performance increase is caused by the decreased necessary confirmation waiting time necessary for transactions [21].

We believe that this makes intuitive sense within the given analogy. Having fewer people involved in controlling a country, speeds up the process in which consensus on policies is achieved.

F. Leased Proof of Stake

Leased Proof of Stake (LPoS) [22] can be seen as the variant between normal PoS and DPoS. The problem with normal PoS, is that for a lot of people their stake is too small to ever stake enough to produce a block. In LPoS, as implemented in Waves [23], becoming a mining node requires a certain amount of tokens. Users that do not have enough tokens to mine themselves, can lease their tokens to one of the miners and share the profits.

The advantage of this protocol is that the participation of the platform users is higher, users with a small stake can also join in the staking process and earn some revenue. Also, just as with DPoS, consensus is more efficient as less nodes have to agree with each other.

G. Proof of Capacity

Proof of Capacity[24], also called Proof of Space, is similar to Proof of Work, but instead uses the allocation of a certain amount of memory as a challenge instead of a computational power to come up with a valid proof. Proof of Capacity was also first created to prevent spam and denial of service attacks. A variations of this consensus protocol are used in for example Burstcoin [25] and Filecoin [26].

How the protocol essentially works, is for a node to add a block to the blockchain, it first has to use some resources and has to prove this to all other nodes in the network. The resource in this case is memory, and the proof is a piece of data with which the other nodes can verify that memory was used. The proof is the solution to a puzzle, just as with Proof of Work, but instead of it being a puzzle which needs CPU power to solve, it needs a large amount of memory.

The biggest upside is that Proof of Capacity uses a lot less energy than a protocol like Proof of Work. Also, memory is almost the same for everyone, whereas hashing power can be increased with the use of a GPU, FPGA or an ASIC, instead of a CPU.

H. Proof of Activity

Proof of activity [27] is a consensus protocol proposed for Bitcoin based on a combination of Proof of Work and Proof of Stake. It was created in order to mitigate some of the problems that each of the proofs has, such as the centralization of PoW pools due to miners only caring about making money off of the most profitable cryptocurrency.

PoA works by having every miner try and solve an empty block header consisting of the hash of the previous block, the miner's public address, the height relative to the genesis block and a nonce using PoW. When a miner succeeds, the empty header block is broadcast to the network, which is then used to pseudo randomly choose N $satoshis$ (smallest unit of currency recorded in the bitcoin Blockchain, it has a value of 100 milionth of a bitcoin). Then these $satoshis$ are followed until the address of their current stakeholders are found. If a stakeholder finds that they are in possession of one or more of these N $satoshis$, they sign the hash of the empty header block with the private key that controls the $satoshi$ and broadcast the signature to the network. When the N^{th} stakeholder finds that one of the $satoshis$ belongs to them (all the other $N-1$ have already signed the empty block header), they collect transactions, sign a wrapped block with all the data and then broadcast it.

By $following\ the\ satoshi$, PoA makes it so stakeholders with more stake are chosen with a higher probability, as it is more likely that they own the $satoshi$ resulting from the hash of the empty header block.

If one of the N stakeholders is offline, the empty header block that selected them will go into a deadlock, but, since other miners are also trying to solve the block, which will result in a selection of N different stakeholders, to overall result is that everything will keep going as normal. And it also serves as an incentive to have more stakeholders active.

I. Proof of Burn

Proof of Burn is a consensus algorithm similar to proof of stake. The main difference is in the way tokens are 'sacrificed' to participate in block creating. In Proof of Stake, tokens are 'staked', wherein the more a user stake, the greater the probability is that they will be creating the new block. In Proof of Burn, tokens are not staked, they are 'burned', made unusable. Tokens are made unusable by transferring them to addresses that do not have a corresponding private key. An address / public key without a private key is unable to move the token, rendering them useless. The currency used to destroy could be either the coin's own token, or other crypto currencies such as Bitcoin.

Slimcoin [28] is an implementation of Proof of Burn. It was built as an alternative to Proof of Stake blockchains.

J. Proof of Elapsed Time

Proof of Elapsed Time (PoET) [29] is a consensus algorithm proposed by Intel based on its trusted computing platform SGX, and is the consensus algorithm of Hyperledger sawtooth[30].

PoET works by having each node generate a random number to determine how long it has to wait before being allowed to generate a block, this is based on a distribution specified by the system in advance. When a new block is submitted, SGX helps the node generate a proof of the waiting time, which can be easily verified by other nodes with SGX technology using a statistical test to determine whether the waiting time lies where specified in the distribution mentioned before.

When compared to PoW, PoET allows new blocks to be created with less computing power, as each participating node does not need to perform expensive computations.

K. Implicit consensus

The implicit consensus protocol is a theoretical consensus protocol. It was theorized in 2017 in the paper 'Implicit Consensus: Blockchain with Unbounded Throughput' by Ren, Cong, Pouwelse and Erkin [31]. A real implementation which uses this consensus model is the Trustchain [32] developed at TU Delft.

This protocol works in a system in which every node has its own individual blockchain, which only records transactions related to itself. These blockchains contain two types of blocks, Transaction Blocks (TB), which are similar to the ones in Bitcoin, and there are CheckPoint Blocks (CP). These CPs contain no transactions, instead they contain an already established consensus and a hash of the previous block.

Implicit consensus works in rounds, whenever a round is done the next consensus round starts. In the next round, each node generates a consensus message (CM) containing the digest of the following items: the node, the round number, the digest of its latest CP, its position in the blockchain and the position of the CP before that one in the blockchain. These items are encrypted using the private key of the node.

After these rounds, an existing BFT algorithm is used to reach agreement on a set of input CMs of

this round, and the result. This result is a vector consisting of the CMs ordered by their node, its output. If a node has a CP in this result, it shall generate a CP with this consensus and append it to its chain.

The only downside of using this protocol is that it requires all nodes to be connected at all times. If every user always has a running node, it would allow for infinite scalability. The throughput in a group of nodes is only dependent on the amount of the nodes forming it. All nodes are independent from one another in terms of throughput, as they only have to add their own transactions to their blockchain.

L. Consensus in a Tangle

Due to the limitations of blockchain, a different approach is being implemented by the cryptocurrency IOTA. IOTA is being developed without incorporating blockchain and replacing it with a structure called a tangle[33].

Compared to the better known cryptocurrencies, where the transactions are stored in the blockchain, a tangle maintains a distributed ledger in the form of a Directed Acyclic Graph (DAG).

In a tangle, every issued transaction forms a node - called site - in the DAG, which then has to approve two previous transactions. These approvals are represented by directed edges, from the approver to the approved, an unverified site is called a tip. If two sites A and B, are not directly connected by an edge, but there is a path of at least length 2 from A to B, we say that A $indirectly$ $approves$ B. These approvals go deeper and deeper into the DAG, until they reach the "genesis" transaction. The genesis transaction is the beginning of the tangle, which contained all the tokens (no tokens are created after this transaction) and distributed them to several "founder" addresses.

The main idea behind a tangle is that by submitting a transaction, you are forced to approve two existing transactions, thereby contributing to the network's security. As a transaction receives more approvals, it receives a higher level of confidence from the system. A higher level of confidence makes it hard for a double-spending transaction to be accepted. When approving its two selected transactions, the node must solve a cryptographic puzzle similar to those in Bitcoin.

When a transaction is issued, it is given a weight proportional to the amount of work the issuing node invested into it. This node can only have a value which is a power of 3. The higher this weight is, the more "important" this transaction is. A transaction will also receive the weight of all transactions that directly or indirectly approve it. This makes older transactions more important than recent ones.

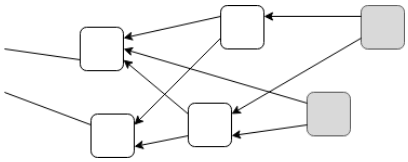

Figure 2: The tangle. White squares represent verified sites, gray squares represent tips.

IV. Different scalability techniques

The scalability of the blockchain is vital to the global adoption and use of these platforms. In the following sections the different techniques to scale a blockchain will be discussed. From these sections, it will be clear that if we want scalability, we have to make a trade off in either decentralization or security. Vitalik Buterin, the creator of Ethereum, calls this the scalability trilemma [34].

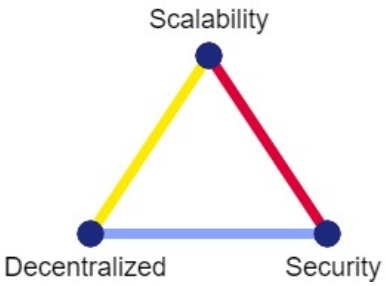

Figure 3: The trilemma occuring in blockchains; only two out of 3 characteristics can be chosen.

A. Parameter tweaking

As has been shown in the sections about Proof of Work and Proof of Stake, many blockchains have a certain speed, which they try to maintain for every new block to be appended to the blockchain. For Bitcoin this is 10 minutes per block. This speed, together with the amount of transactions per block is what determines the throughput.

A block in Bitcoin is capped at 1 megabyte of space. The amount of bytes for each transaction can differ based on the amount of inputs to this transaction. A basic transaction, with 1 input and 2 outputs is about 250 bytes, and average size is about 700 bytes [35]. Resulting in about 1 MB / 700 bytes = 1428 transactions per block, without accounting for the overhead in a block. Another scalability solution would to increase the block size. If we would for example, increase the block size to 10 mega byte, we would instantly be able to handle 10 times more transactions.

A fork of Bitcoin called Bitcoin Cash made a copy of the Bitcoin blockchain, and increased the block size to 8 mega byte [36]. In the Bitcoin Cash network the amount of transactions, is not as high as for Bitcoin. One MB would be enough. However, we can see the effect of the implementation of the bigger block size.

The trade off that is made by increasing the block size, is that of loss of decentralization. When the block size becomes too big, small nodes with low computer power and storage will drop out of the network. In the worst case, only large companies would be able to run full nodes in the blockchain network, which encourages centralization.

The other parameter that can be tweaked is the block generation speed [37]. If the creation of a proof becomes easier, more blocks can be proposed and appended to the blockchain. Subsequently, the throughput would increase. With lowering the difficulty of the proof, we have to sacrifice on a couple of matters [38]. The bandwith will increase, more forks will occur and the wasted work on blocks which are already mined will increase.

B. *The usage of different blockchains*

A solution to the scalability problem, would be to create different blockchains to handle the transactions in parallel. These can be either copies/forks of a certain blockchain, but can also consist of totally different blockchains. This will be able to handle more transactions, however it will result in a shattered network. The Main trade-off would be made in the security of this method. Since computing power is distributed across multiple blockchains, the chance of attacking a network increases. Moreover, the value of each network decreases as fewer connections between people can be made. If someone wants to send value from one altcoin to another, it will be a hassle, because it first has to be exchanged into the other currency.

C. *Sharding*

Sharding [34] is the process of dividing the network into sub-parts such that every part will verify a portion of the transactions. This way transactions can be done in parallel across multiple shards. Every shard has a separate blockchain. There is a certain division made so transactions from certain addresses, or accounts, are handled and stored. An example is when we have two shards to take the modulo of 2 of an address. The transactions of the even addresses are handled by shard zero, and the uneven transactions are handled by shard one.

The intra-shard transactions are only recorded by the nodes of that shard. Transactions between different shards need to have a special transaction on both sides. The shard where the value leaves, has a deletion transaction, and the shard where is value is received, has a special creation transaction. The usage of shards prevents all transactions from being sent to the whole network. However, more importantly, not every node in the network is working to validate the same set of pending transactions.

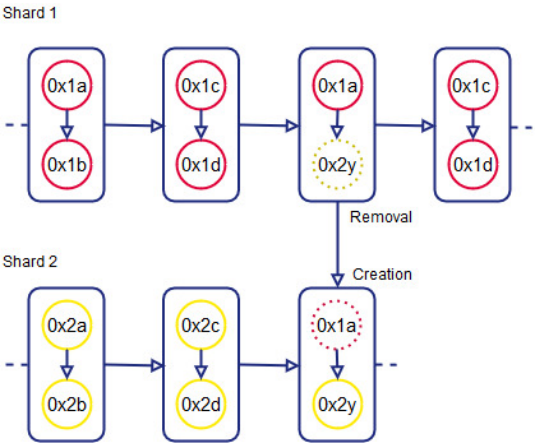

Figure 4: A visualization of sharding, where one transaction is done from shard 1 to shard 2, tokens are removed from 0x1a and created, and added to 0x2y

Does sharding have any trade offs in terms of loss of decentralization or security? If Proof of Work is used as consensus algorithm, it could have security risks. If every node would always work on one shard, and the work is exactly divided between the shards, a security risk will present itself. The amount of computing power that would be needed to attack with the network, is equivalent to the hashing power of one shard. Meaning that if the system has one hundred shards, it would be possible to attack the network with only 1% of the hashing power. To combat this, there should be a randomized algorithm, such that the node is not able to predict what shard they will work in. Ethereum is going to switch to Proof of Stake in combination with sharding. The randomized algorithm for nodes to validate transactions on a certain shard is still needed. The reason to switch is because it is much easier to divide the nodes by their stake instead of their hashing power, as with PoS the incentive to join a mining pool is reduced. [39].

D. *Offchain transactions*

Offchain transactions are a solution to the scalability problem that blockchains, such as bitcoin, struggle with. The solution they provide is creating a channel between two parties, which consolidates all micropayments between them into one larger singular transaction at a later time [40].

Channels work by having both parties agreeing on the balance each of them has when the channel is opened. Then, every time one party wants to send money to the other, they can either update the balance, or create a new transaction which updates the balance. This makes the old transaction invalid. In both cases, both parties can commit to signing the transaction and not broadcasting it. This allows

either party to broadcast the final balance to the blockchain when they see fit.

An application of offchain transactions is the Lightning network [41] which is a large network of channels that allows users to route their payments through a set of channels. This allows them to make almost any micropayment, without having to broadcast the transaction to the blockchain, and creates a near-infinite amount of transactions inside the network, without placing a large burden on the nodes making the transactions.

Lighting network has some problems, a possible problem would be that large corporations that handle a large amount of transactions, have many channels to their customers. Since everyone has channels to only certain large nodes, we would lose the decentralization of the network, and give more power to the large companies.

V. Comparison of different projects

First, a small introduction to each of the projects is given, then they are compared by placing them besides each other in a table.

Bitcoin
Bitcoin (also called Bitcoin Core) has been the first fully decentralized digital currency. It was created in 2008 by a developer (or group) with the pseudonym Satoshi Nakamoto. At this moment, it is still the largest currency in terms of market cap and hashing power. Bitcoin is also the most well known cryptocurrency.

Bitcoin Lightning
Bitcoin Lightning is actually the Bitcoin Core blockchain, but with the possibility to settle transactions instantly off-chain via channels between two users. This would make transactions very cheap and very quick, but only if a path exists from you to the receiver/sender.

Bitcoin Cash
Bitcoin Cash is a hard fork of the Bitcoin blockchain, and has been brought to life to lower the scalability problems and high transaction fees. The only change of this project with respect to the Bitcoin, is that the block size is reduced to 8 mb from 1 mb.

Ethereum
While Bitcoin only focuses on creating a decentralized currency, Ethereum is working hard to become the platform which makes it easier for developers to create decentralized applications. Similar to the Bitcoin blockchain, Ethereum uses Proof of Work. However, the performance is improved by other factors. These factors include the manner at which transactions are stored in the blockchain. Ethereum, unlike Bitcoin, support smart contracts. These smart contracts are the backbone for creating decentralized applications [42].

Ethereum with Sharding
Sharding is the way Ethereum will make their blockchain more scalable. Ethereum has not released a version with Sharding yet, but when it will come out, transactions per second will increase dramatically.

EOS
EOS markets itself as the 'Ethereum Killer'. It seeks to become a richer and more efficient platform for decentralized applications. EOS also includes a number of additional features, such as; grouped permissions for EOS accounts, integration with the Interplanetary File System and an elaborate governance system [43].

Bitshares
Bitshares is a decentralized cryptocurrency exchange. The inner workings of Bitshares are similar to that of EOS. This is the case because EOS is based on Bitshares and Steem. The three blockchains mentioned here, are all made by the same engineer; Daniel Larimer. [44]

Waves
Waves [23] focuses on the custom token creation on their blockchain. It should be easy to create your own token, and add it to their decentralized exchange Tidex. Currently 12,670 different tokens [45] have been released on Waves, although the market cap of these custom tokens is not near the custom tokens on the Ethereum blockchain. The Waves blockchain uses Leased Proof of Stake, with the so-called Bitcoin-NG protocol for extra scalability. Together their protocol is called Waves-NG.

Ripple
It could be a hassle for banks to send funds from an account in one bank to an account in another bank. Especially between banks which reside on the other side of the world. Ripple is a project to connect these banks, and make transactions between them cheap and fast.

IOTA
IOTA is a cryptocurrency which abandons the blockchain format and replaces it with a DAG. DAGs aim to improves confirmation times and network security. By using a DAG, IOTA is more resilient to the advances in quantum computing than the current blockchain implementations.

Burstcoin
Burstcoin [46] is a cryptocurrency which has been set up to fix the flaws which are in the large Proof of Work blockchains. Burstcoin uses Proof of Capacity. The resource which is used to be able to add a new block to the blockchain is memory, instead of computational power at Proof of Work. This way Burstcoin makes a more energy efficient, (possibly) faster and more decentralized platform.

BigchainDB
BigchainDB is a database with blockchain characteristics. Most of the scalability methods are trying to expand the current blockchains be able to work with large amounts of data/transactions. BigchainDB tries to solve the scalability the other way around. They start with the big data solutions which are already there, and are really scalable, to which they try to implement the blockchain characteristics, like decentralization and immutability.

Tribler
Tribler [47] is a peer-to-peer client to anonymously share files. Not only files can be shared like is possible with torrents, Tribler also implements the Trustchain to give users incentive to cooperate on the network. Uploading files is the way to earn seeding tokens, and you can spend them to speed up downloading a file.

Tendermint
Tendermint [18] is a framework used to create Byzantine Fault Tolerant blockchains and decentralized applications. It was first released in 2015. It uses the Proof of Stake consensus algorithm.

Hyperledger
The Hyperledger Project [48] is a collaborative effort to build a framework for open source distributed ledgers. The Hyperledger project was established in 2016 by the Linux Foundation. One of its cornerstones is Hyperledger Fabric, which is built using the Hyperledger standards. This platform can be used to create business solutions for problems, that need confidentiality and a rigor permission system governing the ledgers. While other platforms focus on a purely transparent and permissionless system, Hyperledger believes confidentiality and permissions are necessary for many practical business solutions.
The Hyperledger "Sawtooth" makes use the consensus algorithm Proof of Elapsed Time [30], while maintaining similar goals to Hyperledger Fabric, which uses PBFT.

Cardano
Cardano is a platform used to create decentralized applications. Cardano uses the Ouroboros protocol. Ouroboros is an implementation of Proof of Stake, as mentioned in its corresponding section of this paper. Cardano markets itself as the most advanced smart contract protocol, built in a research driven manner. [49]

Dash
Dash is the cryptocurrency focused on simplifying payments between merchants and consumers. Because Dash has masternodes that verify transactions along with nodes that mine, it is able to provide quick transactions and privacy.

Zilliqa
Zilliqa [50] calls itself 'the next-gen high throughput blockchain platform'. The main focus of their platform is also to create a scalable blockchain platform. Just like Ethereum is implementing, it will use sharding to make it scalable. For consensus it will use PBFT, which as they say stood the test of time, and ensures finality in transactions.

Byteball
Byteball [51] is a general purpose cryptocurrency using a Directed Acyclic Graph to store its transactions. Besides being a currency, it offers smart contracts, custom tokens, sovereign identity, prediction markets and the possibility to have untraceable tokens called blackbytes.

Peercoin
Peercoin [52] solves part of the energy problem of Bitcoin by implementing a hybrid consensus model which has Proof of Work, and Proof of Stake. The Proof of Stake protocol is a variation of the general PoS, and also adds coin age instead of only the size of the stake.

A. Table with projects

Table I on page 9 shows scalability characteristics and metadata about each project. The table is ordered based on the year the project is founded. The columns 'Theoretical throughput' and 'Algorithmic complexity' are most important in reasoning how scalable a blockchain is. The metadata added are in the columns 'Founded', 'Deployed', 'Consensus algorithm' and 'Market cap'. 'Founded' was added to see if a development in scalability can be seen over the years. 'Deployed' is added because it could have an influence on the theoretical throughput. After seeing peak performance in practice, the theoretical throughput can be changed, which should be taken into account

Project	Founded	Deployed	Consensus algorithm	Theoretical throughput (tx/s)	Algorithmic complexity	Market cap ($)	Sources
Bitcoin	2009	Yes	Proof of Work	7	$\mathcal{O}(1)$	146,304,424,888	[15]
Ripple	2012	Yes	Variation of PBFT	1500	$\mathcal{O}(1)$	24.844.146.883	[53], [14]
Peercoin	2012	Yes	Hybrid PoW and PoS	10	$\mathcal{O}(1)$	38.530.630	[52]
Bitshares	2014	Yes	Delegated Proof of Stake	3300	$\mathcal{O}(1)$	428.841.437	[44], [21]
IOTA	2014	Yes	Tangle	$\mathcal{O}(txs)$	$\mathcal{O}(txs)$	3.607.191.015	[33]
BurstCoin	2014	Yes	Proof of Capacity	4	$\mathcal{O}(1)$	23.729.199	[46]
Ethereum	2015	Yes	Proof of Work	20	$\mathcal{O}(1)$	51,789,793,136	[42]
Dash	2015	Yes	Proof of Work	56	$\mathcal{O}(1)$	2.527.063.671	[54]
BigchainDB	2016	Yes	PBFT	1+ million	$\mathcal{O}(1)$	-	[55]
Waves	2016	Yes	Leased Proof of Stake	100	$\mathcal{O}(1)$	356.583.000	[23], [56]
Byteball	2016	Yes	Tangle	$\mathcal{O}(txs)$	$\mathcal{O}(txs)$	120.109.376	[51], [57]
EOS	2017	No	Delegated Proof of Stake	1+ million	$\mathcal{O}(n)$	5,212,634,519	[43]
Cardano	2017	Yes	Proof of Stake	7	$\mathcal{O}(1)$	4.299.615.743	[49]
Bitcoin Cash	2017	Yes	Proof of Work	61	$\mathcal{O}(1)$	17,104,181,863	[36]
Bitcoin Lightning	2017	Yes	Proof of Work	$\mathcal{O}(n)$	$\mathcal{O}(n)$	146,304,424,888	[15], [41]
Ethereum with Sharding	2017	No	Proof of Stake	$\mathcal{O}(shards)$	$\mathcal{O}(shards)$	51,789,793,136	[42], [34]
Tribler	2017	Yes	Implicit consensus	$\mathcal{O}(n)$	$\mathcal{O}(n)$	-	[32], [47]
Zilliqa	2017	No	PBFT	$\mathcal{O}(n)$	$\mathcal{O}(n)$	306.444.816	[50], [58]

Table I: Comparison between the scalability of different blockchain projects.

for undeployed projects. The consensus algorithm has a very big influence on the throughput. Finally 'Market cap' is added as column to see if bigger projects are also the ones most scalable. The market caps were taken on March of 2018.

VI. Discussion

For a fast global platform with low transaction costs, we would like the blockchain to be fully scalable. Projects using Proof of Work, Proof of Stake, Proof of Activity and Proof of Capacity, are all the least scalable blockchains out there. The only impactful ways to scale such solutions are to tweak the parameters, or to make disconnected copies of the blockchain.

As shown in the table there are some projects which reach a relatively high throughput, like Ripple which uses Practical Byzantine Fault Tolerance. The problem is that these blockchains are not infinitely scalable. Aside from this, it seems that these blockchains compromise more in terms decentralization. Which is not surprising, considering they are built for financial institutions such as banks.

The large platforms using Proof of Work, are also the ones who caused the discussion about scalability. Bitcoin is implementing the lightning network, and Ethereum is implementing sharding, which are both techniques that are potentially much more scalable in terms of throughput of transactions. The lightning network has a potential danger to become too centralized, as everyone would open channels to large organizations. This leaves sharding as an appealing solution. Sharding's potential pitfall is that it is easier to attack a single shard, than a single blockchain containing all transactions. If sharding is implemented such that nodes validating transactions do not know beforehand on which shard they will work, it would be a solution which has high decentralization, security and scalability.

We find that platforms using Delegated Proof of Stake also have the potential to scale well (or infinitely). In DPoS, the elected block producers are the only nodes capable of creating and appending blocks. Only the approval of other block producers is necessary for blocks to be accepted, as apposed to all 50% of computing power as in Proof of Work solutions. This allows for great potential horizontal and vertical scaling potential within the block producers. The compromise here would be that the platform becomes more centralized. To combat this, the block producers are elected by regular nodes. This protects such platforms from the danger of centralization, because malicious block producers will be voted out.

Other suggestions such as the implicit consensus, as implemented in the Trustchain, and the Tangle, as implemented in IOTA, form potential solutions to the scalability problem. Implicit consensus is a very decentralized, secure and scalable technique, however it has one downside: every user has its own blockchain, which has to be accessible and online at all times. The downside to the Tangle is that it only becomes more scalable as more transactions are performed. Thus, it will potentially have a very high

throughput, but only when users perform a large amount transactions. The security is also based on the amount of transactions that are being done in real-time, an attacker only has to outperform the nodes which are verifying transactions a given time.

The authors would say from this outcome that Sharding, Delegated Proof of Stake or the use of Implicit Consensus are the solutions that will probably solve the scalability problem best. The lightning network and the use of a Tangle are good ways to scale blockchain, but have some trade offs. Finally, we find that the other protocols and techniques discussed are not scalable enough.

VII. Conclusion

This paper covers a non-exhaustive list of consensus protocols and methods. We foresee many more proposals for other scalability techniques in the future. Blockchain is still a relatively new concept, ripe for innovation and improvement. Although most monetary valuable platforms at the moment, use Proof of Work, from our comparisons we can conclude that Proof of Work as a consensus protocol, is least scalable. Similar consensus protocols such as, Proof of Stake and Proof of Capacity, while solving other potential problems, do not have the potential to be very scalable. The only way to substantially scale them, is to make the blocks bigger, or the transaction size smaller, or the difficulty of the proofs lower.

The discussion covers the authors thoughts on the results of the survey. The use of Sharding, Delegated Proof of Stake or Implicit Consensus seem to be the scalability solutions with the least compromises.

References

[1] D. Tapscott and A. Tapscott, *Blockchain revolution: how the technology behind bitcoin is changing money, business, and the world*. Portfolio/Penguin, 2016.

[2] Airbnb: Vacation rentals, homes, experiences & places. [Online]. Available: https://www.airbnb.com/

[3] Uber. [Online]. Available: https://www.uber.com/

[4] Taskrabbit connects you to safe and reliable help in your neighborhood. [Online]. Available: https://www.taskrabbit.com/

[5] Steemit.com.

[6] Bitcoin average transaction fees. [Online]. Available: https://bitinfocharts.com/comparison/bitcoin-transactionfees.html

[7] Average confirmation time. [Online]. Available: https://blockchain.info/charts/avg-confirmation-time

[8] H. Gilbert and H. Handschuh, "Security analysis of sha-256 and sisters," 2003.

[9] S. T. Aras and V. Kulkarni, "Blockchain and its applications – a detailed survey," 2017.

[10] Unconfirmed bitcoin transactions. [Online]. Available: https://blockchain.info/unconfirmed-transactions

[11] Ethereum blockchain explorer and search. [Online]. Available: https://etherscan.io/

[12] Stress test prepares visanet for the most wonderful time of the year. [Online]. Available: https://www.visa.com/blogarchives/us/2013/10/10/stress-test-prepares-visanet-for-the-most-wonderful-time-of-the-year/index.html

[13] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems*, vol. 4, no. 3, p. 382–401, 1982.

[14] M. Castro and B. Liskov, "Practical byzantine fault tolerance," *OSDI*, vol. 99, pp. 173–186, 1999.

[15] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system." 2008.

[16] I. Bentov, A. Gabizon, and A. Mizrahi, "Cryptocurrencies without proof of work," 2016.

[17] K. O'Dwyer and D. Malone, "Bitcoin mining and its energy footprint," 2014.

[18] J. Kwon, "Tendermint: Consensus without mining," 2014.

[19] A. Kiayias, I. Konstantinou, A. Russell, B. David, and O. R., "A provably secure proof-of-stake blockchain protocol," 2014.

[20] P. Vasin, "Blackcoin's proof-of-stake protocol v2," 2016.

[21] Delegated proof-of-stake consensus. [Online]. Available: https://bitshares.org/technology/delegated-proof-of-stake-consensus/

[22] Wavesplatform. Blockchain leasing for proof of stake. [Online]. Available: https://blog.wavesplatform.com/blockchain-leasing-for-proof-of-stake-bac5335de049

[23] ——. Waves whitepaper. [Online]. Available: https://blog.wavesplatform.com/waves-whitepaper-164dd6ca6a23

[24] G. Ateniese, I. Bonacina, A. Faonio, and N. Galesi, "Proofs of space: When space is of the essence," *Security and Cryptography for Networks - 9th International Conference*, 2014.

[25] Burstcoin - the green innovative cryptocurrency. [Online]. Available: https://www.burst-coin.org/

[26] Filecoin. [Online]. Available: https://filecoin.io/

[27] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, "Proof of activity: Extending bitcoin's proof of work via proof of stake."

[28] P4Titan, "Slimcoin; a peer-to-peer crypto-currency with proof-of-burn," 2014.

[29] Poet 1.0 specification. [Online]. Available: https://sawtooth.hyperledger.org/docs/core/releases/latest/architecture/poet.html

[30] Hyperledger sawtooth. [Online]. Available: https://sawtooth.hyperledger.org/docs/core/releases/latest/introduction.html

[31] Z. Ren, K. Cong, J. Pouwelse, and Z. Erkin, "Implicit consensus: Blockchain with unbounded throughput." 2017.

[32] P. Otte, M. de Vos, and J. Pouwelse, "Trustchain: A sybil-resistant scalable blockchain," *Future Generation Computer Systems*, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17318988

[33] S. Popov, "The tangle," 2017.

[34] Sharding faq. [Online]. Available: https://github.com/ethereum/wiki/wiki/Sharding-FAQ

[35] Bitcoin average transaction size. [Online]. Available: https://charts.bitcoin.com/chart/transaction-size

[36] Bitcoin cash. [Online]. Available: https://www.bitcoincash.org/

[37] A lower block time could help bitcoin scale, but will it work? [Online]. Available: https://www.coindesk.com/lower-bitcoin-block-time-scale/

[38] V. Buterin. On slow and fast block times - ethereum blog. [Online]. Available: https://blog.ethereum.org/2015/09/14/on-slow-and-fast-block-times/

[39] Sharding faq. [Online]. Available: https://github.com/ethereum/wiki/wiki/Sharding-FAQ

[40] Working with micropayment channels. [Online]. Available: https://bitcoinj.github.io/working-with-micropayments

[41] J. Poon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments." 2016.

[42] A next-generation smart contract and decentralized application platform. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper

[43] Eos.io technical white paper. [Online]. Available: https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md

[44] D. Larimer and F. Schuh, “Bitshares 2.0: Financial smart contract platform,” 2015.
[45] Waves platform. [Online]. Available: https://wavesplatform.com/
[46] S. Gauld, F. von Ancoina, and R. Stadler, “The burst dymaxion an arbitrary scalable, energy efficient and anonymous transaction network based on colored tangles,” 2017. [Online]. Available: http://www.burst-coin.org/wp-content/uploads/2017/07/The-Burst-Dymaxion-1.00.pdf
[47] Tribler - privacy using our tor-inspired onion routing. [Online]. Available: https://www.tribler.org/
[48] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, “Architecture of the hyperledger blockchain fabric.”
[49] Cardano. [Online]. Available: https://www.cardano.org/en/home/
[50] Z. team, “The zilliqa technical whitepaper,” 2017. [Online]. Available: https://docs.zilliqa.com/whitepaper.pdf
[51] A. Churyumov, “Byteball: A decentralized system for storage and transfer of value.” [Online]. Available: https://byteball.org/Byteball.pdf
[52] S. King and S. Nadal, “Ppcoin: Peer-to-peer crypto-currency with proof-of-stake,” 2012. [Online]. Available: https://peercoin.net/assets/paper/peercoin-paper.pdf
[53] D. Schwartz, N. Youngs, A. Britto *et al.*, “The ripple protocol consensus algorithm.” [Online]. Available: https://ripple.com/files/ripple_consensus_whitepaper.pdf
[54] Whitepaper dash. [Online]. Available: https://github.com/dashpay/dash/wiki/Whitepaper
[55] Bigchaindb. [Online]. Available: https://www.bigchaindb.com
[56] Waves-ng stress test: results in! [Online]. Available: https://blog.wavesplatform.com/waves-ng-stress-test-results-in-44090f59bb15
[57] Byteball — smart payments made simple. [Online]. Available: https://byteball.org/
[58] Faq - zilliqa. [Online]. Available: https://www.zilliqa.com/faq.html