Working Paper presented at the

Peer-to-Peer Financial Systems 2017 Workshop

2017

Blockchains: Distributed Consensus Protocols

Gerard Briscoe
UCL

Tomaso Aste
UCL

Powered by

Blockchains: Distributed Consensus Protocols

Abstract

Blockchains are expected to provide disruptive innovations through the radical disintermediation of trust from *distributed consensus*. However, the different protocols for *distributed consensus* in blockchains have yet to be exhaustively documented and critically compared. We identify, define and compare the different protocols of *distributed consensus* for blockchains. We first define Distributed Consensus Blockchains (DCBs) as *distributed ledgers* in which the order of transactions is peer-validated by computationally-mediated *distributed consensus*. We then identify and define the different protocols of *distributed consensus* for blockchains. We then compare the different protocols for *distributed consensus* in terms of resources utilisation for determining consensus, energy use, and *block time* for transaction validation. We then discuss the difficulty of future proofing protocols against resource centralisation, given the development of Application-Specific Integrated Circuits (ASICs) for the Proof-of-Work (PoW) protocol of the Bitcoin DCB. We suggest that the Proof-of-Capacity (PoC) protocol can be future proofed, better than the alternative protocols, against resource centralisation. We conclude by suggesting future work into the transparency implications for DCBs, and their potential for disrupting business models.

Keywords: blockchains; distributed consensus; protocol

1 Introduction

Since the first implementation of a blockchain in 2009, as part of the Bitcoin currency, more than seven hundred blockchains have been created at the time of writing this paper [1]. We consider the most significant property of blockchains, distinguishing them from other forms of distributed computing, is the use of *distributed consensus*. Based upon this *distributed consensus* blockchains are expected to be as significant as automation, computing, robots and the Internet, in contributing immensely to cultures [2]. This is because *distributed consensus* is expected to have considerable potential for disruptive innovation [3], through a radical disintermediation of trust to provide distributed peer-to-peer (P2P) alternatives to current hierarchical institutional arrangements [4]. However, the different protocols for *distributed consensus* in blockchains are yet to be critically documented. So, we identify, define and compare the different protocols of *distributed consensus* for blockchains. We investigate the range of blockchain-based implementations available, and analyse specific examples of the different protocols for *distributed consensus*.

The remainder of this paper is organised as follows. In the next section (2) we define DCBs, and then define the protocols for *distributed consensus* in section (3). We compare the protocols in section (4), and discuss their implications in section (5). We then present our conclusions in section (6).

2 Distributed Consensus Blockchains (DCBs)

A *ledger* is a book, or other collection of financial accounts, containing regularly recorded accounts to which transactions are posted. A *distributed ledger* is digitally replicated and synchronised across multiple nodes, with control over the evolution of the transactions potentially shared. A *blockchain* is a specific type of *distributed ledger* in which the transactions are grouped into units termed blocks [5]. Blockchains were introduced to track ownership of the virtual distributed currency Bitcoin [5]. In blockchains, the blocks of transactions are cryptographically hashed into a continuous chain of cryptographic-hash-based consensus [5]. Consensus in blockchains is the process of agreeing on the order of transactions, grouped as blocks, managed by cryptographic hashing. Cryptographic hashing is used extensively within blockchains. Addresses on blockchains are derived by hashing, for example an address on the blockchain of the Bitcoin currency is computed from the SHA2Ð256 hash function. Collision resistance of hash functions is important to avoid two or more nodes generating the same address (a collision), as both (or more) could use payments sent to that address [6]. Also, cryptographic signatures are fundamental to blockchains for determine which transactions are valid. These signatures are generated from a hash of data to be signed. Blocks in a blockchain are identified by their hash, serving the dual purpose of identification and integrity verification. An identification string that also provides its own integrity is called a self-certifying identifier [6]. Blockchains form a record that can only be changed by re-determining consensus, which can be extremely difficult with *distributed consensus* [5]. We shall focus on blockchains with distributed consensus, termed Distributed Consensus Blockchain (DCB), because we consider that they offer potential for new forms of distributed computing.

Distributed Consensus Blockchains (DCBs) are *distributed ledgers* in which the order of transactions is peer-validated. Each transaction notes the sender and the recipient (or recipients) of payment, which can be native *tokens* (such as a Bitcoin) of the blockchain, for goods, services, or to meet a legal obligation. A peer-to-peer (P2P) network observes and verifies the transactions between peers, which are first broadcast to the network and then verified by computationally-mediated *distributed consensus*. The broadcasted transactions are grouped in blocks, appended in linear fashion to the previous block [7]. Each block cryptographically hashes the preceding block, representing the history of transactions [8]. So, the blocks are then chronologically *chained* together, because the header of each block contains a cryptographic hash that summarises the contents of the previous block, perpetually to the first block of the chain. The sequential chaining of blocks makes fraud or forgery prohibitively difficult, since altering a prior entry would require altering all subsequent blocks [9, 10]. The number of transactions grouped in a block ranges from one (an *empty block*) to an upper bound defined by the *block size*. The time required to create a valid block is termed the *block time*, and varies dependent on the choice of protocol for *distributed consensus*. The design of a blockchain should include incentives for block validation, block broadcast, and the inclusion of transactions into blocks. The blockchain of the Bitcoin currency only incentivises block validation in its original design [5]. There can also be *soft* incentives. For example, including fewer transactions per block would lead to a lower transaction throughput over time, and so devalue the blockchain, which would be undesirable for participants who have invested in block validation.

Only when there is *consensus* that a block is valid, are the included transactions considered verified [11]. Consensus is determined by the *longest* chain created from subsequent blocks appended. The *longest* is the most difficult to create, because it requires the greatest effort (e.g. work). A preceding block is the parent block, and a subsequent block is a child block. For a block on the chain, there can only be one path to the genesis (first) block. However, there can be competing chains from any block [12], representing potential *forks* to two or more divergent paths for the blockchain. Forking is either for the transaction history of the blockchain network, which are termed *soft forks*, or for new rules in transaction validation, termed *hard forks*. One-block forks occur when new blocks are created in parallel, being broadcast to all nodes on the network through a flood protocol [12], and potentially represent competing versions of the transaction history [13]. Eventually, one branch will be longer than the others, with nodes that had not already adopted this branch then switching to this branch. Then, the blockchain fork is resolved and the ledger replicas are consistent up to the genesis block. The blocks discarded by the consensus resolution are called orphan blocks, and the included transactions must then be submitted to new blocks [10]. Blockchains guarantee strong, rather then eventual (weak), consistency [14]. Eventual consistency is a special form of weak consistency in distributed computing, which guarantees that if no new updates are made, eventually all accesses will return the last updated value [15]. Blockchains also guarantee serialisability of the transactions, with a probability that is exponentially decreasing with latency [16]. The history of transactions is never considered definitive, but the most agreed upon. So, there is some exposure to fraudulent double-spending when transactions are first made, with less and less risk as transactions gain confirmations (subsequent blocks), eventually being accepted as *verified in principle* by all nodes within the network [17].

2.1 Permissioning

There are two different approaches for access to blockchains, permissionless which provides open access, and permissioned in which access can be restricted. A permissionless blockchain does not restrict participants providing nodes in contributing to consensus [18]. Every participating node has a replica of the *distributed ledger* and shares responsibility for ensuring it remains current [9]. All permissionless blockchains are public (i.e. the transactions are public), because participating nodes must have access to the transactions in contributing to their consensus of them. They also require *distributed consensus* as anyone may participate in contributing to consensus; i.e. all permissionless blockchains are DCBs. For example, the Bitcoin currency utilises a DCB in which anyone can participate [5].

Alternatively, a blockchain can be permissioned, restricting participants from providing nodes in contributing to consensus [19]. Permissioned blockchains are typically centrally controlled, because a single authority establishes the blockchain and determines who may participate (access control). They can be public or private (i.e. the transactions are private), because only participating nodes must have access to the transactions in contributing to their consensus of them. Permissioned blockchains can be DCBs, and would require *distributed consensus* of the management and addition of new participants. A permissioned DCB would be suitable when blockchains are to be managed by multiple organisations, such as United Nations-lead digital ID networks for people with no official documents [20].

2.2 Smart Contracts

An emerging paradigm with blockchains involves Smart Contracts, computer programmes that automatically execute the terms of a contract. There are other *distributed consensus* approaches, such as Paxos [21] and Raft [22]. However, DCBs are more suited, because they are by definition ledgers for transactions and so natively include mechanisms for transaction processing, which is required for contracts [23]. Furthermore, DCBs enable *distributed consensus* of data contained within transactions, through a strongly agreed upon transparent ledger. Smart Contracts offers *distributed consensus* over computation, based upon the same principles. Smart Contracts are currently logic-based instead of law-based [24], because of the problem of legal enforceability particularly with *permissionless* blockchains [25]. When a preconfigured condition is met, the parties involved can be payed automatically [26, 27]. The concept of Smart Contracts was considered almost two decades earlier [28], but only recently has there been concrete implementations. As blockchains have made it practical to register, verify and execute Smart Contracts [27]. Smart Contracts extend the functionality of blockchains, enabling them to achieve *distributed consensus* over computation [29]. For instance, *distributed consensus* over the result of the execution of a Smart Contract, including changes to balances of accounts, as well as and the data stored within the code of the Smart Contract. The Ethereum currency utilises a DCB to offer Smart Contract functionality, and intends to be a Turing-complete programming language [27]. After meeting certain conditions asset transfer occurs, which would traditionally require contracts created by lawyers and escrow services provided by banks, but could instead be facilitated by Smart Contracts.

3 Distributed Consensus Protocols

The *distributed consensus* protocol in DCBs determines agreed transactions among the nodes of the networks. Specifically, determining the order in which transactions are received, because transaction broadcast across the network may not be instantaneous. Therefore, the order of arriving transactions will not be consistent at all nodes [30]. To achieve *distributed consensus* on the order of transactions, they are timestamped in blocks. The hash of a block includes the hash of the previous block, new transactions, and a time stamp establishes the order of transactions [5]. However, there can be competing blocks with different sets of transactions, and therefore a different ordering of transactions over chains of blocks. The expenditure of effort, in terms of resources (*proof*) or rounds of voting, to create blocks provides a solution for choosing between competing blocks and chains, and therefore agreeing on the order of transactions. The use of different *proof-based* and *voting-based* protocols for *distributed consensus* in blockchains defines them as DCBs.

3.1 Proof-of-Work (PoW)

The term PoW was formalised in 1999 [31], although the concept first appeared in 1992 as a proposal for combatting junk mail (spam) by requiring senders to compute a moderately hard function to send email [32]. PoW protocols achieve *distributed consensus* by applying the computing power of nodes as proof of work in validating blocks of transactions. Nodes act as *validators*, creating candidate blocks from transactions they have received. Each validator hashes the transactions in their candidate blocks, organising them into a *hash tree* with a *root hash*. The *root hash*, the hash of the previous block, the current time (timestamp), and a random number called a nonce are placed into the header of the candidate block. The timestamp makes it more difficult for an adversary to manipulate the block chain, as well as serving as a source of variation for the block header hash. This block header is then hashed to produce the block identifier, which the the PoW protocol requires to be less than a specified threshold value, a target, otherwise it will be rejected by the network when broadcasted. The random number of the nonce is then iteratively incremented to search for a hash below the target. The target is adjusted at regular intervals by the protocol to maintain an average *block time*. After a fixed number of blocks the PoW protocol on each node compares the actual time to generate these blocks against the expected *block time*, modifying the target by the percentage difference. For example, the DCB of the Bitcoin currency adjusts the target every 2016 blocks to maintain an average *block time* of 10 minutes. The successful search for block identifiers is rewarded with the issuance of new tokens (coins) [5], and has become termed *mining*. The term *mining* emerged [33] from the analogy of the search for block identifiers by validators that yields new coins to gold miners digging for gold. While considerable effort is required to find a block identifier, it is trivial to confirm it is valid once found. This is important in ensuring that other nodes can easily confirm the *proof of work* of the block identifier, without having to commit further significant computing resources.

PoW protocols by design incentivise trustworthy participation in block validation through the issuance of new coins for successful block validation. DCBs that use PoW protocols consider the most accurate history of transactions, chain of blocks, to have the greatest exertion of work [8]. This assumes trustworthy nodes control a majority (>50%) of the computing power of the network. So, there is a risk of centralisation of the computing power [34]; i.e. when more that 50% of the computing power is operated within a single country [35]. The PoW protocol of the Bitcoin DCB uses the SHA-256 hash function, which has highlighted the resource intensiveness of the PoW protocol for *distributed consensus*, being estimated to have consumed at least 100 MWh per day [36, 37]. The DCB of the Bitcoin currency has transaction settlement latencies up to the *block time* of 10 minutes, or longer if multiple confirmations (subsequent blocks) are required, which limits usability for applications that require immediate transaction settlement.

Proof-of-useful-Work or useful-Proof-of-Work (uPoW) protocols, protocols are a variation of PoW protocols [38], which is used in the DCB of the Primecoin currency [39] and has a average *block time* of 1 minute. The *useful work* is finding Cunningham chains (sequences) of prime numbers. It involves taking a prime number, doubling it and add one to get another prime number, and continuing until reaching a composite number. The longest known Cunningham chain is of length of nineteen, and it is conjectured and widely believed, but not proven, that there are Cunningham chains of any given length. The Primecoin currency DCB strongly suggests the Proof-of-useful-Work (uPoW) may prove viable for *useful work* of some niche mathematical challenges.

3.2 Proof-of-State (PoS)

PoS protocols were proposed for use in conjunction with PoW protocols on the BitcoinTalk forum in 2011 [40]. It was first implemented as a hybrid PoW/PoS protocol in the blockchain of the PPCoin (now called Peercoin) currency in 2012 [41]. PoS protocols were subsequently developed to be used alone. PoS protocols achieve *distributed consensus* by applying the wealth of nodes, in terms of native tokens (coins) of the blockchain network, as proof of stake in validating blocks of transactions [41]. All the tokens (coins) are created at the beginning, because there is no PoW or other mechanism for their distribution. So, there is an initial coin offering or sale of *pre-mined* coins to handle the initial coin distribution. Nodes act as *validators*, creating candidate blocks from transactions they have received and a *stake transaction* to themselves. This *stake transaction* is their proof of stake in validating a block. Each validator hashes the transactions in their candidate blocks, organising them into a *hash tree* with a *root hash*. The *root hash*, the hash of the previous block, the current time (timestamp), are placed into the header of the candidate block. This block header is then hashed for each new timestamp (every second), and for every coin staked in the *stake transaction*. So, a validator has a probability, proportional to their stake, to validate the block. The PoS protocol requires the hash be less than a specified threshold value, a target, otherwise it will be rejected by the network when broadcasted. This target varies from block to block, and is derived from the target of the previous block, the time to generate the previous block, time since the previous block, and the effective balance of the node; such that nodes with more stake can validate more blocks. The derivation of the target maintains an average block time through gradual adjustments that increases the target to reduce block times, or decreases the target to increase block times. For example, the DCB of the NXT currency maintains a block time of 1 minute. The successful search for a hash below below the target earns the fees associated with the transactions within the validated block, and has become termed *minting* or *forging*. The term minting emerged from the analogy of the hash search to the process of minting coins, because those with the most currency have the greatest authority to mint new coins. The term forging (as a blacksmith rather than fraudulent copying) similarly to minting, as forging can be one of the processes in minting coins. Other nodes can then easily confirm the *proof of stake* hash of the broadcasted block.

PoS protocols avoids the computing power intensiveness, and associated energy usage, of PoW protocols for *distributed consensus*, because they rely on stake rather than computing power. PoS protocols by design assume that having a stake in the system will incentivise trustworthy participation in block validation [34]. It therefore requires that trustworthy nodes control a majority (>50%) of the stake of the network. So, there is risk of centralisation of stake rather than computing power as in PoW protocols; i.e. when more that 50% of the stake is controlled by the greatest stakeholder(s). While DCBs that use PoW protocols consider the most accurate history of transactions (chain of blocks) to have the greatest exertion of work, PoS protocols have no equivalent and so posses the *nothing at stake* problem. This is where consensus risks never resolving [42], because nodes have nothing to lose by validating for multiple blockchain histories. When there are competing forks (chains) from a block, for nodes acting as validators (miners) in PoW protocols the optimal strategy is choosing to validate on the chain with the greatest exertion of work. This is because splitting their mining (computing power) to validate blocks across many chains only reduces the potential rewards (new coins) they could earn. For nodes acting as validators (minters) in PoS protocols the optimal strategy is to validate on every chain. This is because they can use their stake to mint, validate blocks, across multiple chains to claim the reward (transaction fees) no matter which wins. One approach with PoS protocols to the *nothing of stake* problem is for every transaction in a block to contribute to a consumed coin age, which is how long the coins have been held without being used. Then the PoS protocol favours the chain with the highest *consumed coin age*, because the greater the coin age in minting the higher the chance of block validation. The use of coin age in PoS protocols risks encouraging hoarding and discouraging spending, because gaining coin age at a higher rate than others requires holding coins instead of spending (which consumes/resets coin age) [**?**]. This may encourage currencies utilising DCB with PoS protocols that use coin age to be more a *store of value* than a *medium of exchange*. The NXT currency became the first DCB with pure a PoS protocol, and has a block time of 1 minute. So, it has transaction settlement latencies up to 1 minute, or longer if multiple confirmations (subsequent blocks) are required, which limits usability for applications that require immediate transaction settlement.

Delegated (or distributed) Proof-of-Stake (dPoS) protocols are a variation of PoS protocols, which is used in the DCBs of the Crypti currency and has an average *block time* of 1 minute. Each node with stake can delegate block validation to other nodes by voting. However, voter apathy in elections has centralisation risks, which can lead to reduced robustness [3].

3.3 Proof-of-Importance (PoI)

PoI protocols were introduced for consensus in the blockchain of the NEM currency [44]. PoI protocols achieve *distributed consensus* by applying the reputation of nodes, through indicators such as participation in valid transactions, as proof of importance in validating blocks of transactions [41]. All the tokens (coins) are created at the beginning, because there is no PoW or other mechanism for their distribution. So, there is an initial coin offering or sale of *pre-mined* coins to handle the initial coin distribution. Nodes act as *validators*, creating candidate blocks from transactions they have received. Each validator hashes the transactions in their candidate blocks, organising them into a *hash tree* with a *root hash*. The *root hash*, the hash of the previous block, the current time (timestamp), are placed into the header of the candidate block. This block header is then hashed such that a validator has a probability, proportional to their importance, to validate the block. The PoI protocol requires the hash be less than a specified threshold value, a target, otherwise it will be rejected by the network when broadcasted. The derivation of the target maintains an average block time through gradual adjustments that increases the target to reduce block times, or decreases the target to increase block times. For example, the DCB of the NEM currency maintains a block time of 1 minute. The successful search for a hash below below the target earns the fees associated with the transactions within the validated block, and has become termed *mining*. Other nodes can then easily confirm the *proof of importance* hash of the broadcasted block.

PoI protocols also avoid the computing power intensiveness, and associated energy usage, of PoW protocols for *distributed consensus*, because they rely on importance rather than computing power. PoI protocols by design aims to encourage nodes to not simply hold tokens as with PoS, but instead actively carry out transactions. However, PoI protocols reliance on *participation in valid transactions* may encourage participation in transactions for participations sake, simply to increase *importance*. For this reason more sophisticated measures of importance are utilised in determining importance. For example, the DCB of the NEM currency uses the *NCD aware rank*, considers the network clustering of nodes, in combination with the net transfers, expenditure in the last 30 days (the more recent transactions being weighted more) and the vested currency for block validation [44, 46]. PoI protocols by design assume that participation, rather than stake or computing power, in the system will incentivise trustworthy participation in block validation. Also, PoI protocols are not at of 51% attacks (controlling more than 50% of the critical resource of the network), as *importance* cannot be amassed like computer power in PoW or stake in PoS. PoI protocols, similar to PoS protocols, are subject to the *nothing at stake* problem, where consensus risks never resolving [42], because nodes have nothing to lose by validating for multiple blockchain histories. This is because they can use their *importance* to validate blocks across multiple chains to claim the reward (transaction fees) no matter which wins. One approach with PoI protocols to the *nothing of stake* problem is to include immutable blocks periodically, such that nodes do not accept the introduction of different chains. For example, the DCB of the NEM currency every 360 blocks the current chain becomes immutable, guarding against *long range* attacks. These *long range* attacks aim to change the history of DCBs over many blocks, by attackers secretly working on competing chains over those many blocks and then releasing them to other nodes on the network to subvert previous consensus. However, this risks inadvertently supporting *short range* attacks, by encoding them permanently with the blockchain. The the DCB of the NEM currency, which uses a PoI protocol [47], has a block time of 1 minute. So, it has transaction settlement latencies up to 1 minute, or longer if multiple confirmations (subsequent blocks) are required, which limits usability for applications that require immediate transaction settlement.

wikiblockpedia

community digital infrastructure CCC

3.4 Proof-of-Activity (PoA)

PoA is a combination of PoW and PoS, with *activity* emphasising that only active stakeholders get rewarded for the vital services that they provide to the network [48]. PoW based protocols give decision-making power to nodes who perform computational tasks, while PoS based protocols give the decision-making power to nodes who hold stake in the system [48]. In PoA operates like PoW initially, with miners working to solve a cryptographic puzzle. The blocks mined do not contain any transactions, instead the miner's reward address. After which the system switches to PoS, and based on information in the header, a random group of validators are chosen to sign the new block. Therefore, the more coins a validator owns the more likely they will be chosen. Distributed consensus is established through *follow-the-satoshi* (smallest token), which involves picking a pseudorandom stakeholder in a uniform fashion. This is achieved by selecting a pseudorandom index between zero and the total number of satoshis in existence up to the last block. Inspecting the block in which this satoshi was minted and following each transaction that transferred this satoshi to a subsequent address, until reaching the address that currently controls this satoshi [49]. If some of the selected validators are not available to complete the block, then a competing block is selected and a new group of validators are chosen, until a block receives the correct amount of signatures. Fees are then split between the miner and the validators who signed block.

PoW risks centralisation as data centres that are dedicated to computation and transaction verification may, due to economies of scale, be able to outcompete individual miners. While PoS risks centralisation from large stakeholders who may similarly try to exert control over the system. A PoA protocol depends upon both computing power (PoW) and stakeholder proportions (PoS), with the addition of stakeholders being online to participate. Also, the PoW component of the protocol is essential for managing the *nothing at stake* problem [48]. Overall, this mitigates the risks of centralisation from computing power or stake by requiring the centralisation of the both by a single actor, or a set of colluding actors, to control the ledger. However, criticisms of PoA are the same as for both PoW (energy required to mine blocks) and PoS (deterring validators from double signing).

The DCB of the Decred currency (**`decred.org`**) uses PoA (hybrid PoW/PoS) consensus protocol. It utilises miners and minters, aimed at creating a more robust currency [49, 48]. In DCB that use PoW protocols miners, who operate much of the infrastructure, typically have considerable influence, while other nodes have relatively little influence. However, Decred allows all nodes to participate in block validation without expensive mining hardware.

3.5 Proof-of-Burn (PoB)

PoB is the provable destruction of a valuable resources to ensure the integrity of a blockchain [50]. PoB protocols operate by *burning* resources, in which users make these resources unusable. For example, sending tokens verifiably to an address that cannot be redeemed, i.e. to an address which is a hash of a random number (for which the chances for someone picking public and private keys for it are negligible). Burning coins is an expensive process for individuals, which further maintains their scarcity. However, it consumes no resources other than the burned underlying assets, unlike PoW. By burning coins one acquires the rights for life-time mining, chosen in a deterministic (pseudo-random) manner among the owners of burnt coins. Therefore, the more coins burned, the higher the chance of being chosen as the creator of the next block in achieving distributed consensus. This *burning* can be considered as a deposit you cannot get back in PoS, or virtual mining hardware which never degrades in PoW. However, over time stake in the system decays, so eventually users have to burn more coins to increase their odds of being selected as validators. The Slimcoin blockchain, based on Peercoin, makes use of PoB in combination of PoW and PoS.

Depending on the implementation of a PoB protocol, users may burn the native cryptocurrency or the currency of another blockchain. Therefore, PoB can be used for bootstrapping one cryptocurrency from another, for which the valuable resource *burnt* is therefore the more established cryptocurrency. The DCB of the Counterparty currency exchange (`counterparty.io`) involves burning Bitcoins to proportionally generate native tokens, through the colored coins functionality. The PoB protocol is implemented by provably rendering funds unusable for future transactions [51], in which participants transfer Bitcoin to an un-spendable address in exchange for native tokens. The Counterparty tokens were distributed proportionally to everyone who destroyed Bitcoins by sending them to an unrecoverable address during a PoB period [52]. The PoB was used to ensure the legitimate distribution of coins, and helped to establish credibility as the developers do not gain anything from the *burnt* Bitcoins required.

3.6 Proof-of-Deposit (PoD)

In the PoD, also called Proof-of-Validation, protocol have to place a security deposit, for a certain period of time, to earn the right to produce blocks [53]. Blocks have a difficulty proportional or equal to the amount of coins that must be offered for *deposit*, having a known block reward. Deposited coins remain untouchable for some length of time and the block reward is delivered to the miner. Either immediately or over a period of time like a dividend or interest payments. The protocol governs through the controlling of these security deposits, which implicitly governs the incentives of validators. Consensus ends at the validator node itself and nodes are chosen randomly to create a new block. As there is one deposit per block there are a limited number of deposits available per unit time. If deposits are appearing too fast then the return must be too high, so the difficulty is increased (return is lowered) and thus demand decreases.

Validation is based on a chain selection rule called GHOST (Greedy Heaviest Observed Sub Tree) [54]. The advantage of GHOST is a strong convergence of history, which means that every block would either be fully abandoned or fully adopted. GHOST modification is that blocks that are off the main chain can still contribute to the irreversibility of a chain [54]. When a validator validates a transaction that GHOST considers invalid, the validator loses their deposit and forfeits the privilege of participating in the consensus process. This directly addresses the *nothing at stake* problem, since behaving maliciously is now costly [53].

The Tendermint blockchain uses the PoD protocol [55, 56], in which validating blocks requires depositing coins in a time-locked bond account, during which they cannot be moved.

3.7 Proof-of-Capacity (PoC)

PoC (also called proof-of-space) protocols require the allocation of hard drive space, generating large data sets by repeatedly hashing a public key and nonces, known as *plots*. This is partially motivated by the observation that users often have significant amounts of free disk space, which is essentially under-utilised [57]. The size of plots determines mining efficiency, as mining involves checking available plots. Every block the node will skim through the saved plots, and afterwards the node can idle until the next block. Thus, the more plots you have the better your chance of finding the next block in the chain.

Since memory access time varies significantly less than processing power in different computing platforms, PoC is likely to be more distributed than PoW. This is because under-utilised storage is more prevalent and than under-utilised computing power (CPUs, GPUs, FPGAs) [58]. This helps to avoid centralised control of a large portion of the network, which could lead to risks such as double spend attacks. It is also more energy efficient than PoW as participants do not have to burn anything. However, there is still the *nothing at stake* problem. As participants can scan their plots to try their chances with a number of alternate chains simultaneously, without spending significant resources. Disk access takes significant time, so it might be more efficient to generate new chunks on the fly, but this would then be akin to simple PoW protocol.

PoC has been used in the Burstcoin blockchain, which it claims to be a green protocol that favours smaller miners by design, making transaction costs cheaper and the network more distributed [59]. Proof-of-useful-Capacity or useful-Proof-of-Capacity (**uPoC!**) protocols is a variation (also called proof-of-retrievability), which is used in the Permacoin blockchain to ensure data preservation. This uPoC is a proof that a target file is intact [60], rather than a useful computation as in uPoW, such that users effectively contribute to a widely distributed and replicated storage system.

3.8 Federated Byzantine Agreement (FBA)

In FBA the consensus mechanism assumes participants knows of other participants, and can distinguish which it considers important. Allowing each node to select a set of other trusted nodes, which induces so-called flexible trust, meaning that all users have the freedom to trust any combination of parties. Nodes may select those participants based on arbitrarily criteria such as repudiation. To find consensus, a node waits for the vast majority of trusted nodes (their quorum slice sets) to agree on a transaction before considering the transaction settled. In turn, those nodes do not agree to the transaction until the participants they consider as important agree to the transaction as well. In turn, the important participants mentioned do not agree to the transaction until the participants they consider important agree as well. Eventually, enough of the network accepts the transaction, such that it becomes infeasible for the attacker to roll it back [61, 62]. So, FBA relies on small overlapping sets of trusted parties. These sets are from nodes that trust each one another. When enough sets of trusted parties are formed, the rest of the system reaches consensus, based on the fact that some trusted parties did. Good behaviour of nodes would ascend nodes into small sets of trusted parties, with the level of trust built over time. Moreover, security depends on digital signatures and hash families whose parameters can realistically be tuned to protect against adversaries with unimaginably vast computing power [61].

Good quorum share nodes and lead to quorums that overlap. We call this overlap quorum intersection. When quorums do not intersect disjoint quorums result. If quorums are disjoint, quorum A can, for example, agree on a statement to order pizza, while quorum B can agree on a statement to order hamburgers. Because they can independently agree on contradictory statements, disjoint quorums can undermine consensus [61]. Each node is responsible for ensuring that its choice of quorum slice does not violate quorum intersection. Making a responsible choice generally boils down to ensuring slices are large enough and that the nodes they contain are important enough not to risk their reputations by lying and feeding different information to different people.

FBA is a majority voting system, related to the decisions of selected trusted nodes. As in every voting system and, especially, if voting nodes are known to each other in a closed system, strategic voting cannot be excluded. It may be possible for a minority of nodes to influence the votes of other nodes, based upon the social structures outside of the overlapping quorum slices, leading to a strong influence of a minority on the voting outcome. So, a minority would influence the majority of nodes to subsequently vote for a particular outcome, i.e. hierarchies in social relationships outside the system could be harnessed to overcome the overlapping quorum slices. Stellar, an open source protocol for value exchange that uses a FBA consensus mechanism [63].

4 Comparison of Protocols

While the PoW protocol remains the most established for DCBs, instances of the PoS protocol are becoming established. Novel alternatives such as PoI, PoB, PoD, and PoC are emerging, as well as hybrids such as PoA. Also, the voting-based FBA protocol is being experimented with in the DCB. Table 1 summarises key aspects of the different protocols for *distributed consensus* for DCBs. Including, the degree of anonymity they offer in their use, the level of immutability of their ledger, and the potential for scalability. Figure 1 is a visualisation of different blockchains, available at **`blockviz.hopto.org`**, showing different blockchains that use PoW protocols.

PoS protocols should be more democratic than PoW protocols, because there should be is a lower risk stake being centralised rather computing power. For PoW protocols influence tends to concentrate with those with the most powerful computers, and not everyone has the wealth to purchase or the skill to maintain such specialist equipment. For PoS protocols validation can be done on any computer, with the investment required in the currency (tokens) of the protocol. Also, the cost of gaining a controlling significant stake for a 51% attack on PoS protocols should be prohibitive.

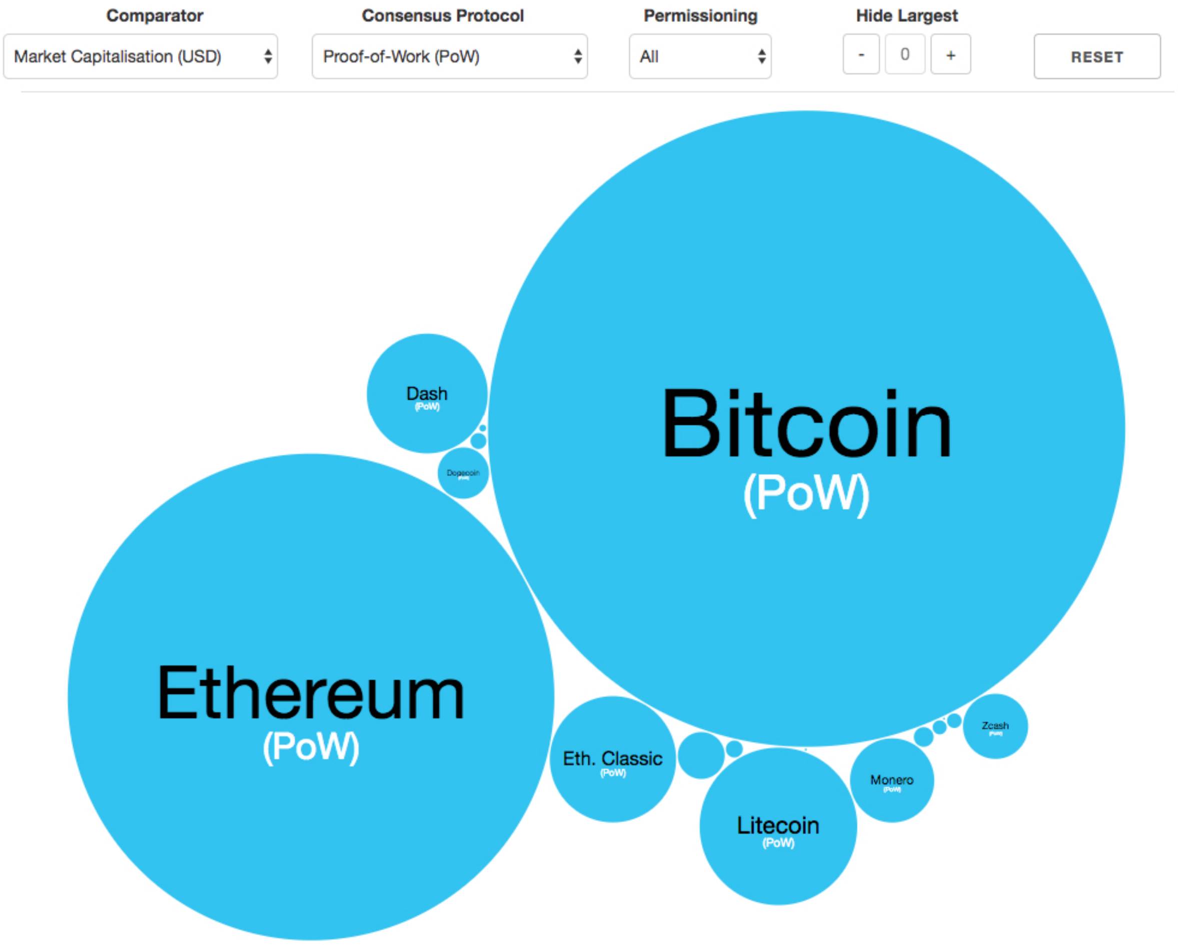
PoI protocols can operate as a hybrid in combination with PoS [45], in which stake can be a pre-requisite for importance in block validation. Alternatively, block validation can iterate between PoI and PoS protocols [45], in which block validation iterates between stake and importance.

5 Discussion

The choice of consensus protocol naturally depends on the intended application, for example if speed for consensus of financial transactions is required, then FBA or PoD may be chosen. However, it also depends on the model of distributed consensus that is desired. Similar to political systems, whether to choose direct democracy through referendums, or representative democracy through the election of officials [64]. Therefore, how democratic is consensus to be achieved, relative to the availability and distribution of the resource to be used for consensus. In other words, what is the risk of centralisation of the resource for consensus. It would appear that the best choice of protocol for *distributed consensus* to minimise the risk of centralisation would be the PoC protocol, because in the FBA protocol a minority may be able to influence the majority to alter data if there is there is an inherent hierarchical trust. However, the same challenge in future proofing the PoC protocol exists, as it did with PoW protocol. While the transition of nodes from Central Processing Units (CPUs) to Graphical Processing Units (GPUs) for mining was to be expected, even to Field Programmable Gate Arrays (FPGAs), the creation of affordable applicable ASICs [65] was likely not expected. We suggest that the unexpected development of ASICs for mining of the PoW protocol of the Bitcoin DCB has led to greater centralisation of the control of the Bitcoin currency than expected. Similarly, for PoC protocols can expect capacity to move from large single drives to high density drives (10TB+), and then to groups of drives (e.g. Network Addressable Storage). However, also similarly, it is difficult to future proof for unexpected innovation and developments in capacity (storage) more generally. Nonetheless, the distribution of capacity as a resource, compared to processing power (work), is greater. Also, the distribution of the speed and size of capacity as a resource, compared to processing power (work), is also currently greater. Ultimately, the best choice may be combine different protocols for *distributed consensus*, so necessitating controlling multiple resources to centralise control. Similar to the PoA protocol, which necessitates controlling two resources, processing power (work) and stake, to take centralise control.

Protocol	Resource	Energy	Example	Block Time	Anonymity	Immutability	Scalability	Notes
PoW	Processing Power (electricity)	High	Bitcoin	10 mins	High	High	Low	Greatest real world use.
uPoW	Processing Power (electricity)	High	Primceoin	1 min	High	High	Low	Difficult to find suitable usefulness.
PoS	Token Ownership	Low	NXT	1 mins	High	Moderate	Moderate	Nothing-at-stake problem.
dPoS	Token Ownership, Peer Reputation	Low	Crypti	1 mins	Moderate	Moderate	Moderate	Centralisation risk from voter apathy.
PoI	Transaction Activity, Token Ownership	Low	NEM	1 min	High	Moderate	Moderate	Vulnerable to Sybil attacks. Nothing-at-stake problem risk.
PoA	Processing Power, Token Ownership, Network Presence	Moderate	Decred	5 mins	High	High	Moderate	Reduces risk of centralisation from a single resource.
PoB	Token Destruction	Low	Counterparty	10 mins	High	Moderate	Moderate	PoS benefits without the drawbacks.
PoD	Token Escrow	Low	Tendermint	1 sec	High	Moderate	Moderate	Nothing-at-stake problem can persist.
PoC	Storage Space	Low	Burstcoin	4 mins	High	High	Moderate	Nothing-at-stake can persist. Can morph into PoW.
uPoC	Storage Space	Low	Permacoin	-	High	High	Moderate	Yet to be a working example.
FBA	Peer Reputation (quorum vote)	Low	Stellar	2-5 secs	Moderate	Moderate	Moderate	Solves nothing-at-stake without PoW.

Table 1: Comparison of Distributed Consensus Protocols: Considering resources utilised, energy use required, and an example with average block time. Also the anonymity, immutability and scalability, as well as notable observations.



Comparator: Market Capitalisation (USD)

Consensus Protocol: Proof-of-Work (PoW)

Permissioning: All

Hide Largest: 0

Figure 1: Figure 1 is a visualisation of different blockchains, available at `http://blockviz.hopto.org`, allowing comparison based upon the protocol for *distributed consensus*. Currently showing blockchains that use PoW protocols.

6 Conclusion

We defined Distributed Consensus Blockchains (DCBs) as permissionless *distributed ledgers* in which the order of transactions is peer-validated by computationally-mediated *distributed consensus*. We then defined the different protocols for *distributed consensus*, for which seven are *proof-based* and one is *voting-based*. We then compared the different protocols for *distributed consensus* in terms of resources utilised for determining consensus, energy use, and block time.

DCBs have been characterised as *trust machines*, since their protocols for report transactions with precision through machine-based *distributed consensus*, rather than verification by human-based third parties [66]. DCBs enable *distributed trust* from transparency mediated by *distributed consensus*, potentially challenging existing intermediaries serving that function and creatively disrupting how they operate [24]. So, future work should consider the transparency implications of DCBs, and any differentiation resulting from the protocols for *distributed consensus*. DCBs are seen to have potential to enable new transnational organisational forms [67], deriving from the trust through transparency that they are expected to afford [68, 9]. So, future work could also consider the potential of DCBs for Business Models (BMs), for both existing BM in payment system networks, and the potential for new BMs in the emerging Blockchain Distributed economy.

DCBs are considered [24] to be important as the Hypertext Transfer Protocol (HTTP) protocol that underlies the World Wide Web. So, we may be at the outset of what could become a fully-fledged Blockchain Distributed Economy with DCBs enabling every aspect of human endeavour. Therefore, the current blockchain-related activity could be seen as early-stage prototypes, which will come to be viewed as primitive data networks at some future moment.

7 Acknowledgments

This work was supported by the Engineering and Physical Sciences Research Council lead Research Councils United Kingdom's Digital Economy Theme, Glass Houses: Transparency and Privacy in Information Economies project, grant EP/N028104/1.

References

[1] Cointelegraph. Explore the visualized history of the cryptocurrencies. MapOfCoins.com; 2017.

[2] Manyika J, Roxburgh C. The great transformer: The impact of the Internet on economic growth and prosperity. McKinsey Global Institute. 2011;1.

[3] Mattila J, et al. The Blockchain Phenomenon–The Disruptive Potential of Distributed Consensus Architectures. The Research Institute of the Finnish Economy; 2016.

[4] Hopf S, Picot A. Crypto-Property and Trustless Peer-to-Peer Transactions: Blockchain as Disruption of Property Rights and Transaction Cost Regimes? Wulfsberg Redlich Moritz. hopf2016;p. 159.

[5] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. bitcoin.org; 2008.

[6] Rogaway P, Shrimpton T. Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In: International Workshop on Fast Software Encryption. Springer; 2004. p. 371–388.

[7] Kant R. Blockchain: The Financial Pandora's Box. The Market Mogul. 2016;.

[8] Rückeshäuser N. Do We Really Want Blockchain-Based Accounting? Decentralized Consensus as Enabler of Management Override of Internal Controls. In: Internationalen Tagung Wirtschaftsinformatik; 2017. p. 16–30.

[9] Yermack D. Corporate governance and blockchains. Review of Finance. 2017;21(1):7–31.

[10] Decker C, Wattenhofer R. Information propagation in the bitcoin network. In: Peer-to-Peer Computing. IEEE; 2013. p. 1–10.

[11] Underwood S. Blockchain beyond bitcoin. Communications of the ACM. 2016;59(11):15–17.

[12] Kehrli J. Blockchain explained. niceideas.ch; 2016.

[13] Spagnuolo M, Maggi F, Zanero S. Bitiodine: Extracting intelligence from the bitcoin network. In: International Conference on Financial Cryptography and Data Security. Springer; 2014. p. 457–468.

[14] Vukolic M. Eventually Returning to Strong Consistency. IEEE Data Eng Bull. 2016;39(1):39–44.

[15] Vogels W. Eventually consistent. Communications of the ACM. 2009;52(1):40–44.

[16] Sirer E. Bitcoin Guarantees Strong, not Eventual, Consistency. hackingdistributedcom. 2016;.

[17] Seibokl S, Samman G. Consensus: Immutable Agreement for the Internet of Value. KPMG; 2016.

[18] Peters GW, Panayi E. Understanding Modern Banking Ledgers through Blockchain Technologies: Future of Transaction Processing and Smart Contracts on the Internet of Money. In: Tasca P, Aste T, Pelizzon L, Perony N, editors. Banking Beyond Banks and Money. Springer; 2016. p. 239–278.

[19] Swanson T. Consensus-as-a-service: a brief report on the emergence of permissioned, distributed ledger systems. R3. 2015;.

[20] Irrera A. Accenture, Microsoft team up on blockchain-based digital ID network. Reuters; 2017.

[21] Lamport L. The part-time parliament. ACM Transactions on Computer Systems (TOCS). 1998;16(2):133–169.

[22] Ongaro D, Ousterhout JK. In Search of an Understandable Consensus Algorithm. In: USENIX Annual Technical Conference; 2014. p. 305–319.

[23] BitFury Group, Garzik J. Public versus Private Blockchains. Bitfury; 2015.

[24] Mougayar W. The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology. John Wiley & Sons; 2016.

[25] Murphy S, Cooper C. Can smart contracts be legally binding contracts? R3 / Norton Rose Fulbright; 2016.

[26] Crosby M, Pattanayak P, Verma S, Kalyanaraman V. Blockchain technology: Beyond bitcoin. Applied Innovation. 2016;2:6–10.

[27] Buterin V. A next-generation smart contract and decentralized application platform. Ethereum; 2014.

[28] Szabo N. The idea of smart contracts. Nick SzaboÂŠs Papers and Concise Tutorials. 1997;.

[29] Delmolino K, Arnett M, Kosba AE, Miller A, Shi E. Step by Step Towards Creating a Safe Smart Contract: Lessons and Insights from a Cryptocurrency Lab. IACR Cryptology ePrint Archive. 2015;2015:460.

[30] Cap CH. Bitcoin—das Open-Source-Geld. HMD Praxis der Wirtschaftsinformatik. 2012;49(1):84–93.

[31] Jakobsson M, Juels A. Proofs of work and bread pudding protocols. In: Secure Information Networks. Springer; 1999. p. 258–272.

[32] Dwork C, Naor M. Pricing via processing or combatting junk mail. In: Annual International Cryptology Conference. Springer; 1992. p. 139–147.

[33] Bjoernsen K. Koblitz Curves and its practical uses in Bitcoin security. University of California, Santa Barbara; 2009.

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

[35] ZHOU C. Is hash power concentration of over 51% in China a problem? news.8btc.com; 2017.

[36] Peck ME. The Bitcoin Arms Race is on! IEEE Spectrum. 2013;50(6):11–13.

[37] O'Dwyer KJ, Malone D. Bitcoin mining and its energy footprint. In: IET Irish Signals & Systems Conference and China-Ireland International Conference on Information and Communications Technologies; 2014. p. 280–285.

[38] Ball M, Rosen A, Sabin M, Vasudevan PN. Proofs of useful work. International Association for Cryptologic Research; 2017.

[39] King S. Primecoin: Cryptocurrency with Prime Number Proof-of-Work. primecoin.io; 2013.

[40] QuantumMechanic. Proof of stake instead of proof of work. BitcoinTalk.org; 2011. Available from: https://bitcointalk.org/index.php?topic=27787.0.

[41] King S, Nadal S. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. self-published paper, August. 2012;19.

[42] Poelstra A, et al. Distributed consensus from proof of stake is impossible. wpsoftware.net; 2014.

[43] NEM. Technical Reference, Version 1.0. NEM; 2015.

[44] Bozic N, Pujolle G, Secci S. A tutorial on blockchain and applications to secure network control-planes. In: Smart Cloud Networks & Systems (SCNS). IEEE; 2016. p. 1–8.

[45] Watanabe H, Fujimura S, Nakadaira A, Miyazaki Y, Akutsu A, Kishigami J. Blockchain contract: Securing a blockchain applied to smart contracts. In: Consumer Electronics (ICCE), 2016 IEEE International Conference on. IEEE; 2016. p. 467–468.

[46] Kobayashi N, Shijo Y. c0ban: a crypto currency is for advertisement and entertainment apps v0. 2. c0bantrade.com; 2016.

[47] Bheemaiah K. Why Business Schools Need to Teach About the Blockchain. Available at SSRN 2596465. 2015;.

[48] Bentov I, Lee C, Mizrahi A, Rosenfeld M. Proof of Activity: Extending Bitcoin's Proof of Work via Proof of Stake. ACM SIGMETRICS Performance Evaluation Review. 2014;42(3):34–37.

[49] Bentov I, Gabizon A, Mizrahi A. Cryptocurrencies without proof of work. In: International Conference on Financial Cryptography and Data Security. Springer; 2016. p. 142–157.

[50] Tschorsch F, Scheuermann B. Bitcoin and beyond: A technical survey on decentralized digital currencies. IEEE Communications Surveys & Tutorials. 2015;18(3):2084–2123.

[51] Florian M, Walter J, Baumgart I. Sybil-resistant pseudonymization and pseudonym change without trusted third parties. In: Proceedings of the 14th ACM Workshop on Privacy in the Electronic Society. ACM; 2015. p. 65–74.

[52] Elendner H, Trimborn S, Ong B, Lee TM, et al. The Cross-Section of Crypto-Currencies as Financial Assets: An Overview. Sonderforschungsbereich 649, Humboldt University, Berlin, Germany; 2016.

[53] Zamfir V. What is Cryptoeconomics. CryptoEconomicon; 2015.

[54] Sompolinsky Y, Zohar A. Accelerating Bitcoin's Transaction Processing. Fast Money Grows on Trees, Not Chains. IACR Cryptology ePrint Archive. 2013;2013(881).

[55] Bonneau J, Miller A, Clark J, Narayanan A, Kroll J, Felten E. Research Perspectives and Challenges for Bitcoin and Cryptocurrencies. Massachusetts Institute of Technology; 2015.

[56] Kwon J. Tendermint: Consensus without mining. URL http://tendermint com/docs/tendermint {_} v04 pdf. 2014;.

[57] Dziembowski S, Faust S, Kolmogorov V, Pietrzak K. Proofs of space. In: Annual Cryptology Conference. Springer; 2015. p. 585–605.

[58] Abadi M, Burrows M, Manasse M, Wobber T. Moderately hard, memory-bound functions. ACM Transactions on Internet Technology (TOIT). 2005;5(2):299–327.

[59] Yuan B, Lin W, McDonnell C. Blockchains and electronic health records. Massachusetts Institute of Technology; 2015.

[60] Bowers KD, Juels A, Oprea A. Proofs of retrievability: Theory and implementation. In: Proceedings of the 2009 ACM workshop on Cloud computing security. ACM; 2009. p. 43–54.

[61] Mazieres D. The stellar consensus protocol: A federated model for internet-level consensus. Stellar Development Foundation. 2015;.

[62] Rubin J. Federated Systems. Massachusetts Institute of Technology; 2015.

[63] Higgins S. Jed McCaleb Talks Stellar's New Protocol for Consensus. CoinDesk; 2015.

[64] Redoano M, Scharf KA. The political economy of policy centralization: direct versus representative democracy. Journal of Public Economics. 2004;88(3):799–817.

[65] Taylor MB. Bitcoin and the age of bespoke silicon. In: Proceedings of the 2013 International Conference on Compilers, Architectures and Synthesis for Embedded Systems. IEEE Press; 2013. p. 16.

[66] The Economist. The trust machine. The Economist. 2015;.

[67] Swan M. Blockchain: Blueprint for a new economy. O'Reilly Media Inc; 2015.

[68] Pilkington M. Blockchain technology: principles and applications. In: Olleros F, Zhegu M, editors. Research Handbook on Digital Transformations. Edward Elgar Publishing; 2016. p. 225–253.