IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 13, 2019, accepted February 27, 2019, date of publication March 21, 2019, date of current version April 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2904181

Applicability and Appropriateness of Distributed Ledgers Consensus Protocols in Public and Private Sectors: A Systematic Review

A. SHAHAAB[1], B. LIDGEY[2], C. HEWAGE[1], AND I. KHAN[1]
[1]Cardiff School of Technologies, Cardiff Metropolitan University, Cardiff, CF5 2YB, U.K.
[2]Companies House, Cardiff, CF14 3UZ, U.K.

Corresponding author: A. Shahaab (ashahaab@cardiffmet.ac.uk)

This work was supported by the Knowledge Economy Skills Scholarships through a major pan-Wales operation supported by the European Social Funds through the Welsh Government, and through Companies House, U.K., under Grant CMK219.

ABSTRACT Advancement of consensus protocols in recent years has enabled distributed ledger technologies (DLTs) to find its application and value in sectors beyond cryptocurrencies. Here we reviewed 66 known consensus protocols and classified them into philosophical and architectural categories, also providing a visual representation. As a case study, we focus on the public sector and highlighted potential protocols. We have also listed these protocols against basic features and sector preference in a tabular format to facilitate selection. We argue that no protocol is a silver bullet, therefore should be selected carefully, considering the sector requirements and environment.

INDEX TERMS Blockchain technology, consensus algorithms, distributed consensus protocols, distributed ledger technology, DLTs for public sector, distributed systems, Govtech, permissioned and permissionless blockchains.

I. INTRODUCTION

Achieving consensus is a fundamental problem in distributed computing. Lamport *et al.* [1] discussed the challenges of achieving consensus in a distributed environment over three decades ago. The authors exemplify the challenge as Byzantine General Problem, where a consensus within the n generals (referred as nodes) is required to establish trust on the information they receive and based on the information decide whether they should attack or retreat from a siege they are currently under. Addressing the Byzantine General Problem and achieving fault tolerance is at the core of any distributed ledger. Fault tolerance is achieved by introducing redundancy and information is agreed upon via agreed consensus protocol. Building resilient systems that can deliver high availability and consistency through redundancy, tolerating network and communication failures, power cuts or any other catastrophe, have been an area of active research for the last three decades. ISIS [2], Paxos [3], Raft [4], view-stamped replication (VSR) [5], Chubby [6] and Zookeeper [7] are just to name a few of the attempts at synchronizing critical information across a distributed ledger. The objective of all of these attempts is to achieve availability and consistency though redundancy.

A consensus protocol must possess three key properties [8]:

1) Safety/Consistency – Given some input, all nodes in a distributed setting should produce the same output.
2) Liveness – The majority honest nodes should keep the network alive.
3) Fault Tolerance – The network should tolerate some faults (f) in a setup of nodes (n).

Fischer *et al.* [9] argue that no deterministic consensus protocol can simultaneously guarantee property A, B & C in a distributed system (FLP theorem). Fault tolerance being critical, distributed systems tend to choose between safety and liveness, depending on the system requirements and set assumptions. Majority of the earlier protocols assumed that the replication environments was trusted and free of adversaries. Distributed Ledger Technology (DLT), however, requires the network to achieve consensus without any intermediaries and in a byzantine environment (Network with adversaries present). Satoshi Nakamoto, through his/her/their landmark paper ‘‘Bitcoin: A Peer-to-Peer Electronic Cash

The associate editor coordinating the review of this manuscript and approving it for publication was Chien-Ming Chen.

System'' [10] have excited the world by the ingenious idea of having a decentralised currency which does not need a central authority. A network of nodes maintain a ledger of all the transactions and they all share the same version of truth via the novel use of PoW, [11] to achieve consensus among the participating nodes. DLTs in general and blockchain in particular, have gained popularity, following its successful implantation in Bitcoin.

Consensus is at the core of any DLT. We reviewed 66 traditional and contemporary consensus protocols that we have come across in the academic literature or whitepaper publications. Based on their characteristics, we have categorised them into architectural and philosophical categories. We then present a comparison of these protocols from business use case point of view and use public sector as a reference and provide a table summarizing our findings for the discussed consensus protocols. We believe that private/permissioned DLTs and off-chain solutions better suit public services sector needs as it offers privacy and control for governmental and other public sector organizations.

We have not found any previous work with such large coverage of consensus protocols and providing any form of visual and tabular representation on how they are interconnected or categorised. None of the earlier attempts discuss the suitability of consensus protocols from public sector point of view or recommended any consensus protocols by analyzing the features of consensus protocol. Previous works include providing frameworks for evaluation or reviewing some of the popular ones. Bach *et al.* [12] has done a comparative analysis of consensus protocols used in top 10 cryptocurrencies by market cap. The authors have focused only on the cryptocurrencies and have not focused on private DLTs. Cachin and Vukolic [13] have scanned the literature on consensus protocols for permissioned DLTs. The focus of writing is on fault tolerance and resilience in permissioned setting. More recent work from Nguyen and Kim [14] is a literature review of several consensus protocols, summarizing and classifying them as proof based and vote based consensus protocols. The authors argue that vote-based consensus protocols are more suitable for private DLTs whereas proof-based consensus protocols are more suitable for public blockchains. Jun [15] has reviewed the landscape of blockchain adoption by governments and have provided summarized tables of different public sector initiatives. Reference [15] goes on tabulating different electronic voting systems and the underlying blockchain technologies. Reference [16] highlights the potential promises and benefits of using DLTs in public sector. Tuan *et al.* [17] have compared some consensus protocols from the network setting point of view, discussing their suitability for public or private DLTs. Xu *et al.* [18] has ranked different blockchain models for cost efficiency, performance and flexibility. Bano *et al.* [19] have reviewed the performance and security of different consensus protocols, providing a common evaluation framework to visualize the capabilities of the protocols.

The rest of the paper is set as following; Section II defines DLT and discusses the variants of DLTs. Section III provides a visual representation of classification of consensus protocols based on the underlying data structure and their suitability for public and private DLTs. Reviewed consensus protocols are also discussed in section III. Section IV briefly discusses the preferred DLT choices of public sector and provides a list of suitable consensus protocols for public sector following the discussion. We then provide our conclusion in section V.

II. DISTRIBUTED LEDGER TECHNOLOGY (DLT)

For non-technical person, DLT is like a WhatsApp group chat. Once a message is sent to the group, the whole group become the witness of ''what, who, when'' of the message. As long as the majority of participants in the group chat are honest, this message will be safe and deemed as truth. From technical point of view, DLT is an approach for maintaining distributed copies of a single ledger across multiple data stores. It allows to record, share and sync data across the network in such a way that the whole network reaches consensus on the content of the ledger and secures the information, such that it cannot be altered in the future. This immutability property of the DLTs make them suitable for a variety of businesses applications where accurate and honest record of historical transactions is important and data sharing between multiple participants is required. Thus DLT finds its use in finance [20], public sector [15], [16], [21], [22], identity management [23], supply chain [24], [25], insurance [26], healthcare [27], [28], IOT [29], [30] and several other domains [29].

Philosophically DLTs can be classified into three broad categories, Public, Private and Consortium, based on the consensus participation, read/write permissions and the level of centralization. Public DLTs are fully decentralised. No one controls the network and participation in consensus process is open to everyone and all transactions are visible to the public. This ''openness'' ensures that the data on the DLT cannot be changed once it has been validated and accepted by the network. Bitcoin and Ethereum are examples of a public DLTs.

In private DLTs, only authorized nodes from an organization can take part in the consensus and have read/write permission in private DLTs. One or multiple entities control the access to the private DLTs, restricting the participation in the network. Hyperledger Fabric and Multichain are examples of private DLTs.

Consortiums are essentially private DLTs shared between multiple organizations. Different organizations come together to form a consortium and nominate members to take part in the consensus process. Quoram and Corda are examples of consortium DLTs.

Private (including consortiums) DLTs offer better finality (stable consensus) because the whole network generally follows the leader and consensus is collaborative

TABLE 1. Comparison of three types of DLTs inspired by [31] and [14].

	Public	Private	Consortium
Consensus Process	Permissionless	Permissioned	Permissioned
Centralization	None (In theory)	Full	Partial
Participation	High	Low	Low
Write	Anyone/Delegated	Pre-selected	Pre-selected
Read	Public	Public/Restricted	Public/Restricted
Efficiency	Low	High	High
Security	Nearly Tamper proof	Can be tempered	Can be tempered
Trust Among Participants	No	Yes	Yes
Finality	No	Yes	Yes
Incentive	Yes	No/maybe	Not/Maybe

(through committee) rather than competitive. Table 1 shows the comparison of the three categories of DLTs

Architecturally, DLTs are categorised into two broad categories – the linear Blockchain and Direct Acyclic Graphs (DAGs). The consensus mechanism in blockchain is competitive type (only one block can make into the blockchain at a time) while in DAGs it is swarm type (transactions/blocks can be added to the network in parallel) [32]. BlockDag is a hybrid of Blockchain and DAG, proposed as a solution to the scalability challenges of blockchain [33].

A. BLOCKCHAIN

Originally used by Haber [34] for secure timestamping of digital documents and later modified by Satoshi Nakamoto in 2008 for the crypto currency Bitcoin [10], blockchain is essentially a peer to peer distributed, immutable, append-only data structure. Transactions are grouped together into blocks and each block has a pointer to its previous block, such that any anomaly in the ''chain'' formation can be easily detected. Nodes maintaining the blockchain agree on the data, transactions, ordering of the blocks and provide a distributed log of events. Blockchain can tolerate a variety of faults and is designed to operate in extreme byzantine environments. Nodes maintaining the blockchain have to contribute to the security of network by investing in computational power, storage, and memory or have direct stake in the platform.

B. DIRECT ACYCLIC GRAPHS (DAGS)

A DAG is a finite directed graph with no directed cycles, consisting of finite number of edges and vertices, where each edge is directed from one vertex to another, such that there is no path that connects a vertex V to itself [32]. Unlike blockchains with competitive consensus models, the consensus model in DAGs is cooperative. DAGs also offer parallelism by allowing a more general connectivity to the existing events stored in the ledger [35].

C. BLOCKDAG

BlockDAG is a hybrid of Blockchain and DAG that offers scalability to the blockchain by mining blocks in parallel and growing the chain in a DAG formation. Non-conflicting transactions are allowed to be mined on separate chains and are selectively merged by mining a block that adds both of these to the main chain [36].

III. CONSENSUS PROTOCOLS

In a distributed environment, consensus protocols are implemented to ensure that all state replications happen according to pre-defined state transitions and rules. Achieving consensus in a distributed system is challenging. Consensus protocols must be resilient to nodes failure, network partitions, message delays, ordering and corruption [37]. Numerous protocols have been proposed, with each protocol making the required set of assumptions in terms of synchrony, message broadcasts, failures, malicious nodes, performance and security of the messages exchanged [37]. Each consensus protocol tries to achieve the stability in the network of $n$nodes where f nodes can be faulty. Generally a network needs $n \geq 2f + 1$ entities to tolerate$f$failures [38].

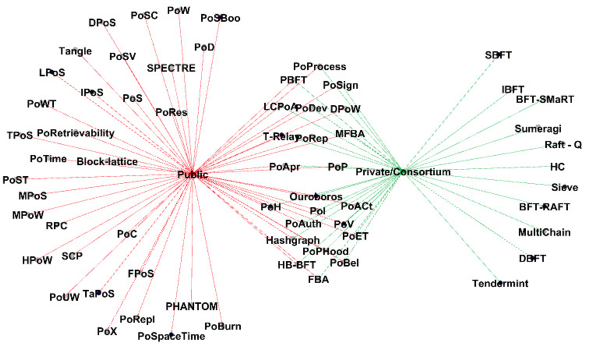

FIGURE 1. Relationship of different consensus protocols with different philosophy-based categories of DLTs. Public category related protocols are linked by red lines while Private/Consortium categories by green lines.

We present the 66 known (at the time of writing this review) consensus protocols for DLTs in Fig. 1 and 2; followed by brief description of each protocol. Fig. 1 presents the protocols in relation to philosophical categorization of DLTs and other attributes listed in Table 1. For visualization ease, we have combined private & consortium DLTs together so that protocols that can be used in all three categories (e.g. PBFT) can be placed in the middle.

Fig. 2 presents the protocols in relation to architectural categorization of DLTs as described in section II. We further describe the consensus protocols, where 1-61 describe the consensus protocols used in blockchain architectures. 62-64 in DAGs and 65-66 in BlockDAGs. Blockchain centric protocols are ordered according to their relevance - starting from the common ones such as PoW, PoS, PoA, PBFT and then discussing their variations.

A. Proof of Work (POW)

Originally proposed by Dwork and Naor [39] to combat phishing emails, Satoshi Nakamoto adopted PoW as a consensus mechanism for Bitcoin [10]. PoW requires the miners

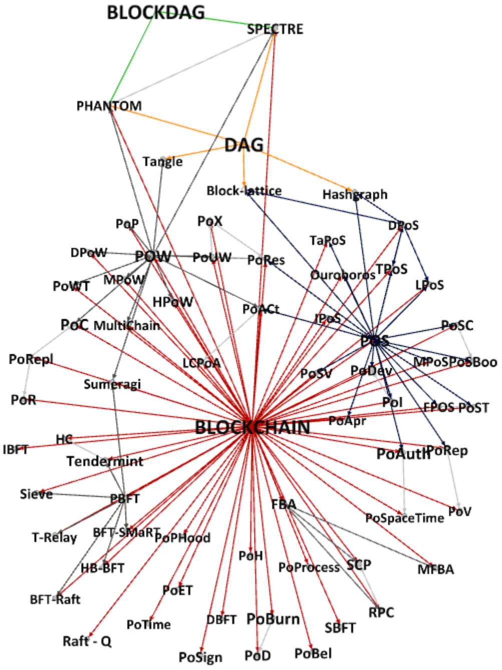

FIGURE 2. **2. Relationship of different consensus protocols with different architecture-based categories of DLTs. Blockchain related protocols are linked by red lines, DAG by orange lines and BlockDAG by green lines. Interrelationship among protocols particularly with common protocols (e.g. PoW, PoS) to others are linked by black lines with saturation representing the number of interrelationships.**

(nodes attempting to add a block to the blockchain) to rigorously find a nonce n which satisfies a difficulty level l, such that combined hash of nonce n and the hash of block header b is less than the set difficulty level. Mathematically l can be written as,

$$H(n||H(b)) < l$$

When such nonce is found, the miner creates the block and announces it to the network. Other nodes in the network then verify the block by computing the hash and verifying the requirements.

Changing anything from a block is impossible without redoing the work. Changing history is even harder as a user will be required to re-compute$n$ that satisfies l for all blocks mined after the block under attack. This requires significant amount of computational power, known as hash rate [10].

B. DELAYED PROOF OF WORK (DPOW)

Newly formed blockchains that do not have enough computational or staking power behind them and therefore are an easy target for attackers. DPoW proposes the use of established blockchain with high hash rate (currently Bitcoin), to secure the transactions on a smaller blockchain which do not have enough computational power behind it. n number of notary nodes are elected using a stake-weighted vote. These notary nodes are responsible for ''archiving'' the data on the selected PoW blockchain. DPoW does not strictly follow the longest chain rule but the longest chain rule is applied up to the most recent backup onto the PoW network. Furthermore, Notaries can elect to switch to another PoW network if the alternative offers greater hashing power or the transaction costs go substantially high [40].

C. PROOF OF PROOF (POP)

Similar to DPoW, PoP aims to enable a security inhering blockchain *IB* (low hashrate or newly formed blockchains) inherit the security of a security provider blockchain *SP* (established blockchains). The miners in PoP publish the current state of *IB* onto *SP* [41].

D. HYBRID PROOF OF WORK (HPOW)

HPoW is an energy considerate variant of PoW consensus protocol. It removes the profit incentive for miners, making it impractical for the mining farms to mine a network using HPoW, encouraging the solo miners with low computational resources to take part in the consensus. HPoW in Lynx requires that a miner cannot have been the recipient of mining reward in previous 60 blocks, the reward address should have a minimum of 1000 coinage (a product of coins in miner rewards address and the difficulty of previous 10th block) and the last two characters of SHA256 hash of miner's reward address must match the last two characters of block hash value [42]. This randomises the winning node and does not guarantee the fastest node to claim the reward.

E. PROOF OF ELAPSED TIME (POET)

Originally proposed by Intel, PoET is intended to run in a Trusted Execution Environment (TEE), such as Intel's Software Guard Extensions (SGX). Block leaders are randomly chosen by lottery-based model of SGX, to finalize the block. Each validating node requests TEE for a random waiting time. All nodes work on the puzzle and announce the block after the waiting time, along with the waiting proof created by TEE that all participant nodes can verify easily [43]. This takes away the advantage of having higher computational power as the miner with smallest waiting time would be able to announce the block quickly. PoET requires dedicated hardware which limits the participation and decentralization. Milutinovic *et al.* [44] has discussed a similar approach called ''Proof of Luck'' where nodes are given a random lucky number instead of waiting time.

F. PROOF OF EXERCISE (POX)

PoX is an extension of PoW based on the idea that the ''work'' done by miner should be useful. Miners are given a computation-expensive, real world matrix-based scientific problem. An Employer E store the ''exercise'' X on available store and deposits some credit on the blockchain, guaranteeing the availability of the exercise. Miner M collects all valid transactions and is assigned a random X. M promises to solve X and deposits a credit on the blockchain as a commitment to solve X. Once M finds the solution Y for X, M creates a verification transaction and publishes it for verifiers V. Once predefined number of V validate the results, M collects all transaction details (exercise, deal, verification and audit) and adds the block to the blockchain [45].

G. PROOF OF USEFUL WORK (POUW)

Similar to PoX, PoUW is an extension of PoW where miners are required to solve a meaningful difficulty. It requires miners to solve Orthogonal Vectors, 3SUM, All-Pairs Shortest Path, and any problem that reduces to them. Delegators post the problems to public problem board and the miners grab those problems to mine the block and attach the proof of useful work to the block. Verifiers check the block by checking the hash of the newly proposed block and that the problem $P(f, x)$ has not been previously solved [46].

H. PROOF OF RESEARCH (PORES)

Proof of research (PoRes) is used in Gridcoin which contributes 6.16 petaFlops [47] to Berkeley Open Infrastructure for Network Computing (BOINC) [48]. PoRes combines PoS and PoW where blockchain is secured using PoS and miner are rewarded separately for performing computations to solve scientific problems. The BIONC project server stores and distributes project data to nodes running BIONC client. Upon completion, the nodes return the results to the server which rewards the node in BIONC credits. These BIONC credits are converted to gridcoins to reward the participants. Similar to mining pools [49], miners are rewarded for their relative processing contribution to the project.

I. PROOF OF WORK TIME (POWT)

In Bitcoin's PoW, difficulty level is regularly adjusted to create blocks at regular interval. This leads to waste of computational power required to find the nonce that satisfies the target difficulty. PoWT proposes a variable block creation rate that scales with the mining power. Block creation rate increases with the mining power, increasing scalability and transaction speed while simultaneously reducing the ''waste'' of computation power required to find the nonce. [50].[1]

J. MAGI'S PROOF OF WORK (MPOW)

The MPoW employs a network dependent reward which limits the network's hash rate. Reward is continuously adjusted based on an attraction repulsion model. The network increments rewards to stimulate network activities during passive mining phase and decrements rewarding to mitigate redundant mining sources during aggressive mining phase [51]. This makes the mining unsuitable for mining pools and allows low end devices to take part in mining. However, it opens the network for an adversary to overcome the network hashing power and launch a 51% attack[1].

K. PROOF OF STAKE (POS)

Proposed for Peercoin [52], Proof of stake is the most popular alternative of proof of work. It does not require the nodes to consume excessive power to secure the network and achieve consensus but rely on the nodes staking their coins to propose the blocks and secure the network. The chances of selection for creating next block depend on a mix of the tokens a node hold and coin age (how long the tokens have been held). The block proposer is required to stake its coin age to append to the blockchain. The stake of the node is slashed if it acts maliciously. Once the validator claims the reward, the coin age is destroyed, allowing others to ''win the raffle''.

L. DELEGATED PROOF OF STAKE (DPOS)

DPoS is similar to PoS, but instead of stakeholders creating and validating the blocks, they nominate N number of witnesses to do it on their behalf. Each witness is randomly chosen to create next block and all witnesses get their turn. The witnesses are reshuffled again once every witness has had its turn. Delegates are also nominated by the stakeholders, which moderate the blockchain and can control the blockchain parameters such as rewards, block intervals, block size, etc [53]. Stakeholders however, are given a cooling off period to react to the changes made and either accept or nullify the changes. Stakeholders can also remove delegates. The transactions are performed much faster compared to PoW or PoS because only selected witnesses participate in the block validation.

M. FAIR PROOF OF STAKE (FPOS)

PoS allows the nodes with higher stakes to create more blocks in comparison to the rest of network. FPoS is a proposed ''improvement'' on PoS, adding a ''fair'' probability of creating a block. Proposed solution is to use exponential distribution instead of uniform distribution in random variable selection [54].

N. INTERACTIVE PROOF OF STAKE (IPOS)

IPoS requires communication among T participants for block generation. The blockchain starts with T genesis blocks instead of one in order to avoid breaking the ticket generation rules. A unique *seed* value (known to all participants) from the block headers determine the generators of next block. A single node is allowed to sign and broadcast a block, but each block required T weighted tickets from all T accounts. Tickets are generated by using a special formula that uses the *seed* value from current and previous block headers, public keys and balance of the accounts. Every block is given a score based on the ticket scores and the block and block and blockchain with highest score wins.

IPoS proposes to minimize the number of variables a miner can iterate over, as a protection mechanism for grinding attacks. No delays or timestamps are explicitly stated and the protocol operates as a weighted lottery where multiple winners create a block [55].

O. PROOF OF STAKE BOO (POSBOO)

PoSBoo is a PoS scheme based on PoS Casper. A set of pre-selected master nodes take part in consensus and block creation [56]. Block reward is a multiplication of fixed block

[1] 51% attack is an event where an adversary controlling more than 50% of network computational power can create his own version of the blockchain history or prevent transactions for gaining confirmations.

reward and network weight. 25% of the staked coins are burned if a node tries to fork the chain by voting for two blocks at the same height. Further penalties are imposed on the nodes voting on false block more than N number of times. There is not much information available about PoSBoo at the time of writing this paper.

P. LEASED PROOF OF STAKE (LPOS)

LPoS is an extension to the PoS consensus protocol. It allows the users to "lease" their balances to other nodes. Nodes with higher number of leased balances have higher chances of selection to produce next block. This reduces the likelihood of network being controlled by a single group of nodes, by increasing the number of electable members [29]. Rewards are shared between miners and lenders.

Q. PROOF OF STAKE TIME (POST)

PoST uses a non-linear proof function that accepts the distribution enhancing time and reject the time that diminishes it, at a given block. This is achieved via a periodic time-acceptance function that is proportional to the coins held and relative to network strength. Idle-time is defined as the fraction of age that no longer supports the distribution of consensus and instead begins to degrade it. This quantified idle-time is unique to each stake, as it decreases the probability to meet the proof and impacts the fraction of earnable matured interest via consensus.

Time active fraction f is mathematically defined as

$$f = cos^2 (\pi p) \{if (p > 0.45), f = m\}$$

where the fraction of accepted age f is equal to the squared cosine of the product of π and that transaction's consensus power p. That is if p is greater than 0.45, otherwise all age is lost, and time active fraction is set to minimum stake time of 8 hours. p is measured as the fraction of coinage of the network wide stake weight time. In order to maximize the probability of earning all matured interest and signing a block during a period of time, a node must stake actively to ensure passage through the Stake-Time window for all coins held [57].

R. PROOF OF STAKE VELOCITY (POSV)

PoSV encourages users to both stake and spend the tokens by using an exponential decay function for coinage. In contrast to the traditional PoS protocols which consider coin age as a linear product, new coins get coin age quickly in PoSV and old coins age slowly. Probability of a node to be selected as block leader depends on the wallet size and wallet activity [58].

S. PROOF OF STAKE CASPER (POSC)

PoSC has been proposed as an alternative to PoW for Ethereum. It was an early attempt at "nothing at stake" problem where validators are penalized for malicious activities. PoSC relies on the *checkpoint* blocks whose height is exact multiples of 100 in the *checkpoint* tree. Validators are divided into dynasties, defined as the number of finalized *checkpoints* from genesis block to the parent of the block. A validator V can join the dynasty $d + 2$ when his deposit is included in the block at dynasty d and can only leave at $d + 2$ dynasty if withdrawing at a block at dynasty d [59].

T. MAGI'S PROOF OF STAKE (MPOS)

MPoS is also designed on the same attraction repulsion models, as MPoW. The stake weight is conditionally proportional to the age and amount of the coin. Stake weight does not always increase with the increase in the coin count and offline staking is limited to a maximum of seven days [60].

U. TRANSACTION AS PROOF OF STAKE (TAPOS)

TaPoS requires all transactions to carry their proof of validity with them, implicitly making all nodes generating transactions to contribute to network security. Every transaction contains the hash of most recent block informing the network that the user's stake is on a particular fork [61].

V. TRUSTLESS PROOF OF STAKE (TPOS)

TPoS allows users to safely stake their offline coins from cold storage. Account *owner* can grant permission to a different address, *merchant* to stake on account holder's behalf. *Merchant* nodes does not take part in block creation or convince nodes to accept transactions but can only validate transactions. Stakeholders meeting the minimum collateral requirements run *masternodes*. *Masternodes* verify transactions, take part in voting and block generation [62].

W. OUROBORO

Ouroboros is a PoS variant which operates in *epoch*, comprising of fixed time *slots*. Slot leaders are elected from the group of "qualifying" stake holders. Each *epoch* has exactly one slot leader who is responsible for creating the block. The slot leaders for $N+1$ *epoch* are elected during *epochN*, hence the network already knows who will be the slot leaders for next *epoch*. The chances of being elected as block leader are proportional to the stake of a node [63].

X. PROOF OF AUTHORITY (POAUTH)

In PoAuth, preselected set of trusted "authorities" are given the rights to propose blocks. The identity of the authorities is verified both online and in public sector. Time is divided into *steps* and each *step* S has a mining leader which can create blocks. For each *step*, authorities take turn on round-robin basis to propose the blocks and a block is accepted onto the blockchain once it has been signed off by majority of the authorized nodes [64]. PoA becomes intrinsically centralized by identifying the authorities. Therefore, it is best suited for private blockchains and consortiums.

Y. PROOF OF REPUTATION (POREP)

PoRep is an extension of PoAuth where the validator nodes are selected based on their reputation. Reputation must be important enough that the participant should face serious consequences financially and brand wise, if they act maliciously. Once the validators are selected, the network then operates as a PoAuth network. Block leader is selected by round-robin lookup and a node can only sign a block every $(N/2)+1$ blocks, given N validators [65].

Z. PROOF OF PERSONHOOD (POPHOOD)

PoPHood makes use of ring signatures [66] and collective signing [67]. A set of volunteer organizers arrange a *pseudonym party* where attending parties are known but individuals can remain anonymous. Each party is given exactly one cryptographic identity, binding their physical and virtual identity. The attendees who want to become *minters* are given a week to authenticate themselves and form a *minting pool*. RandHound [68] is used to generate randomness which is used to select the next block proposer [69].

1) MULTICHAIN

Multichain, a private blockchain, restricts the mining to a set of identifiable identities. A constraint is applied to the number of blocks a miner can produce in a given window, stopping the monopolization of mining process. This implements a round-robin block creation schedule enforced by the *mining diversity* parameter [70].

2) PROOF OF SIGNATURE (POSIGN)

Developed by XTRABYTESTM, PoSign relies on the authorized *STATIC* nodes that communicate on a VPN like network called *VITALS*. A *PULSE* signal is sent to each node whenever a transaction occurs on the network, alerting them to validate and sign the new transaction. Communicating over *VITALS*, online *STATIC* nodes validate and sign each block and are rewarded in transaction fees [71]. Offline nodes will double check the blocks when they come online but they do not sign the blocks.

3) PROOF OF APPROVAL (POAPR)

In Proof of Approval, blocks are published periodically at a pre-defined interval. Any node can propose a "*candidate block*" and broadcast it to the network, however, a stake holder is given weighted privilege. A quorum of stake holders scores the candidate block by checking how close to the target timeslot it was received by them. The nodes will reject the proposed block if it did not include any potential valid transactions or include any invalid ones. The nodes then rank the qualified candidate blocks in descending score and broadcast the list to the network. The creator of candidate block with good score then packs the received approvals, creates an approval block and broadcast it to the network. Both the approval blocks and winning candidate block form the blocks are finally added to the blockchain [72].

4) PROOF OF BELIEVABILITY (POBEL)

PoBel is a variation of PoS that relies on the "believability" score of a node, which is calculated at the beginning of an *epoch*. Each user is given a score called "*servi*", for their long term added value to the community. Believability is a measure of *servi* and stake. *servi* is zerod upon block creation, giving the next node with highest believability score a chance to create the block. Validating nodes are divided into two groups, *believable league* and a *normal league*. PoBel has two phases. In the first phase, a believable validator quickly processes the transactions and proposes a block by validating and ordering a set of committed transactions. In second phase, normal validators sample and verify the transactions. The user loses all its stake and reputation if the normal validator detects any misbehavior [73].

5) PROOF OF IMPORTANCE (POI)

PoI is a variation of PoS where each account is given an importance score based on their stake in the network and the overall network support [74]. Block proposer is selected by choosing a user meeting the minimum stake requirements, who has transferred some funds in last X days and have a rank (NCDawareRank in Nem's case) computed based on stationary probability distribution of Ergodic Markov chain [75].

6) PROOF OF DEVOTION (PODEV)

Proof of devotion is a hybrid of PoS and PoI where accounts with highest influence in the ecology and liquidity are selected. These accounts are given equal rights to create blocks. The top ranked accounts voluntarily stake to become block *validators*. Block proposer is chosen pseudo randomly from the *validators* set. *Validator* sets are divided into *dynasties* and *validators* cannot change *dynasties* within an *epoch* of X blocks. All *validators* from the *dynasty* participate in the round of BFT style, time bound voting to create the block [76].

7) PROOF OF VALUE (POV)

PoV is a spinoff of PoRep, enabling peers to reach a consensus about perceived value of contribution of an individual to a network. Backfeed [77] and Sapien [78] are using PoV to reward positive journalism whereas AI Crypto [79] is using PoV to reward members for the derived value from the projects.

8) PROOF OF ACTIVITY (POACT)

PoAct is a hybrid of PoW and PoS. Empty block header is mined by the miners and the hash of the newly minded block header is used to deterministically choose N pseudorandom stake holders. Each stakeholder checks the validity of the newly mined block template. Upon validation, the first $N-1$ stakeholders sign the hash of the empty block header and broadcast their signature to the network. Nth stakeholder wraps the block by adding as many transactions as it wishes, along with the previously acquired $N-1$ signatures, signs the block with its signature and broadcasts the wrapped block to the network. All nodes check the validation of the block and the block is added to the blockchain. Transaction fees are shared between the miners and N stakeholders [80].

9) LIMITED CONFIDENCE PROOF OF ACTIVITY (LCPOA)

LCPoA is an extension of Proof of Activity where the system creates automatic checkpoints in the blockchain, limiting the possibility of rewriting the history of the blockchain [81]. A 51% attack can still be carried out, but the attacker would only be able to rewrite only a small number of blocks.

10) PROOF OF CAPACITY (POC)

PoC relies on the node's storage capacity instead of the computational power. Miners invest in disk space instead of computing power and dedicating more disk space increases the probability of successfully mining a block. Miners create the chunks of data, known as *plots*, where pre-computed hashes to forge the block are stored. The more plots a node have, the better are the chances of append the next block to the chain [82].

11) PROOF OF RETRIEVABILITY (POR)

PoR works similar to PoC. A prover P is required to store some large dataset F and prove to a verifier V that the P possesses F and F is fully retrievable. This verification takes place as a challenge response protocol where V issues random challenges C and P provides responses R which V can verify without possessing R. PoR allows the network to perform as a decentralised distributed cloud storage [83].

12) PROOF OF SPACETIME (POSPACETIME)

PoSpaceTime is a variant of proof of storage which allows a verifier to verify that a prover has stored its data for some period. The prover generates short sequential proofs of storage by using zk-SNARKS [84] and a verifier can easily verify without interacting with the prover [85] Storage miners put a collateral deposit and commit to store client's data. Miners then generate PoST and submit to the network, as a proof that they are storing the data for agreed time [85].

13) PROOF OF REPLICATION (POREPL)

In PoRepl, a prover P is required to commit to store n physically independent copies of some data D and store D in a dedicated storage. Phas to convince a verifier V that P is storing the unique physical copies instead of duplicating multiple copies of D in the same storage space [86].

14) PRACTICAL BYZANTINE FAULT TOLERANCE (PBFT)

PBFT implements a state machine replication and can tolerate $(n-1)/3$ faults [8]. Network comprises of leader and validating peer nodes. Block creation happens in rounds. Peers receive the transactions, validate them and broadcast to the network. At the end of each round, the leader orders the transactions and put them in a block. Block creation process is categorised as *pre-prepares*, *prepare* and *commit* phases. The leader broadcasts the proposed block to the peers, in *pre-prepare* phase. The peers store the block locally and broadcast the same block to other peers in the *prepare* and *commit* phase. Upon receiving 2/3 validations from the peers, nodes will execute the *commit* phase and add the block to their current blockchain.

15) TENDERMINT

Tendermint is a variant of PBFT, based on DLS protocol [87]. All transactions are first broadcasted to a group of validators, which have some stake locked in the system. The validator nodes vote on the valid transactions for their inclusion in the blockchain. Voting takes place in three steps, *prevote*, *precommit* and *commit*. A block is committed upon receiving 2/3 signatures from validator nodes. Block proposer is chosen in a round-robin fashion, with a proportion to their voting power, i.e. Stake. Tendermint is resilient up to 1/3 of byzantine participants [88].

16) SUMERAGI

Heavily inspired by [89], Sumeragi applies the concept of global order and divides the nodes into two sets, set 1 consisting of $2f + 1$ nodes and set two consisting of the remaining. Considering only $2f + 1$ signatures are required to confirm a transaction, only nodes from set A take part in consensus. Consensus is performed on every transaction in Sumeragi. A lead validating peer verifies the transaction, orders, signs and broadcasts transactions to the remaining validating peers. Other validating peers validate the signature of transaction along with the contents and temporarily update the ledger. It then signs the Merkle root and hash of the transaction's content and broadcasts the finite ordered list of transactions. Nodes keep sharing the valid parts of Merkle tree until roots match [90].

17) THRESHOLD RELAY (T-RELAY)

DFINITY [91] uses a beacon as the source of leader selection and ranking based on the threshold relay technique. A group of nodes called committee is selected to act as notary and derive the random beacon which is used to select the *committee* for next round. A fresh, verifiable random value is produced by the randomness beacon at the beginning of round r. Each node is given a priority rank by the randomness beacon. Any node can pool the transactions and propose a block, but the block proposed by highest priority rank has more chances to be notarized. Upon receiving the blocks, the notary waits for the *blocktime*, ranks the blocks, signs and broadcast the block with highest rank. All nodes then append their copies of the blockchain [91].

18) BYZANTINE FAULT TOLERANCE –SMART (BFT-SMART)

BFT-SMaRt is the only known project that was developed before the interest in permissioned blockchains surged around 2015 [70]. Bessani *et al.* [92] started work on it in 2009. There is widespread agreement today that BFT-SMaRt is the most advanced and most widely tested implementation of a BFT consensus protocol available. BFT-SMART supports a configuration parameter that, if activated, makes the system strictly crash fault-tolerant (CFT). When this feature is active, the system tolerates f < n/2 (simple minority).

Experiments have demonstrated that it can reach a throughput of about 80,000 transactions per second in a LAN with 1000 nodes [92].

19) BFT RAFT– TANGAROA (BFT-RAFT)

Inspired from Raft [4] and PBFT [8], BFT-Raft aims to maintain Raft's safety, liveness and fault tolerance properties. Nodes and users share the public keys with each other ahead of time. Messages are always signed by both nodes and users and messages carrying invalid signatures are rejected. A node can be a *leader*, *follower* or a *candidate*. *Leader* is elected by voting and it serves as a leader for a fixed time term. BFT-Raft network of n nodes can tolerate f byzantine failures where $n \geq 3f + 1$ [93].

20) DELEGATED BYZANTINE FAULT TOLERANCE (DBFT)

DBFT works similar to DPoS. Instead of *witnesses* and *delegates*, DBFT is composed of *ordinary nodes* and *bookkeepers*. The *ordinary nodes* vote for *bookkeepers* and the successful *bookkeepers* take part in the consensus on behalf of the ordinary nodes. A random *bookkeeper* is selected to propose the next block and the block is added to the chain only if more than 66% of the *bookkeepers* agree that the transactions are valid.

The DBFT provides fault tolerance of $n \geq 3f + 1$ [94]. Transaction throughput of nearly 1000 transactions per second (TPS) has been recorded in NEO blockchain with block interval of 15-20 seconds [94].

21) HYDRACHAIN (HC)

HydraChain is an extension of the Ethereum platform which adds support for creating Permissioned blockchains. Inspired by Tendermint, HC consensus protocol is a BFT protocol that relies on a set of validators which form quorums and validate the order of transactions. The block proposer is randomly chosen from the set of validators. Consensus is achieved via one or more rounds on the proposed block and new round can only be started once more than 2/3 nodes have voted on the previous round [95].

22) HONEYBADGER BFT (HB-BFT)

HB-BFT is the first practical asynchronous BFT protocol which does not make any timing assumptions [96]. HB-BFT's design is optimized for scenarios where network bandwidth is scarce, but computation is fairly ample. Consensus is achieved through N number of pre-selected nodes with known identities. The goal of the nodes is to agree on the ordering of the input, given some transactions. The nodes maintain a transaction buffer and store the received transactions in their buffers. The protocol proceeds in *epochs*. At the start of each *epoch*, nodes choose a subset of the transactions from their buffers and provide them as input to an instance of a randomized agreement protocol. At the end of the agreement process, the final set of transactions for the *epoch* is chosen and this new set of transactions is added to the committed log. A throughput exceeding 20,000 transactions per second for networks of up to 40 nodes have been reported by Miller *et al.* [96].

23) ISTANBUL BFT

Istanbul BFT is inspired by PBFT [8] and is used in QuorumChain [97] which is an enterprise focused version of Ethereum. Block proposer is selected randomly from the validators in a round-robin fashion. Newly proposed block is broadcasted to the network with the pre-prepare message. Validators enter the pre-prepared stage and broadcast the prepare message. The block proposer enters prepared state upon receiving $2f + 1$ prepare messages from the validators and broadcasts commit message with a proposal to insert the prepared block to the blockchain. Validators insert the block to their chains upon receiving $2f + 1$ commit messages. Istanbul BFT can tolerate $3f + 1$ faulty nodes in a network of N validators.

24) SCALABLE BFT (SBFT)

SBFT is a parallelization scheme enabling BFT systems to scale with the number of available cores by binding all messages and tasks to a particular processor core. Actors executing the replication protocol are organized in *pillars* where each *pillar* is responsible for certain instances of consensus, executed by a dedicated thread. *Pillar* numbers are kept in direct alignment with the number of cores and requests are managed at the same *pillar* level [98].

25) FEDERATED BYZANTINE AGREEMENT (FBA)

FBA can be considered as the most novel solution to the byzantine general problem. Each participant maintains a list of important nodes that it trusts. A transaction is considered settled when majority of the trusted nodes agree on the settlement. The trusted nodes only consider a transaction settled when the nodes they trust agree on the transaction. Eventually, majority of the network agrees on the transaction, making it immutable [99]. Nodes decide who they can trust. Quorums and slices emerge because of the selections made by the nodes. Ripple and Stellar use their own versions of FBA, both are discussed later.

26) RIPPLE CONSENSUS PROTOCOL (RCP)

Ripple relies on a trusted set of validating nodes to maintains its ledger. The ledger has two forms, *last-closed ledger* and *open ledger*. Each validating node maintains its trusted set of nodes called *Unique Node List* (*UNL*) where each node must have an overlap with other nodes in the Ripple network. Each node collects the latest transactions into a "*candidate set*" and broadcasts its candidate set to the *UNL*. *UNL* nodes validate the transactions and broadcast their votes to the network. Voting takes place in rounds. Transactions that do not acquire validation votes are discarded from the *candidate set*and the *candidate set*receiving over 80% votes from the *UNL* is considered valid and added to the "Last closed Ledger" of Ripple network. Unverified transactions are kept in the *open ledger* until they meet 80% verification target [100].

a: Stellar Consensus Protocol (SCP)

SCP is a variant of FBA and uses the same notion of *quorums* and *quorum slices* instead of trusting the whole network. SCP relies on a set of *validator nodes* to achieve consensus. *Quorum* is a set of nodes adequate to reach consensus and a *quorum* slice is a subset of *quorum* which can help in convincing a node about the agreement. *Quorum* intersections are required in order to achieve broader consensus and finality [99].

b: Modified Federated Byzantine Agreement (MFBA

MFBA is a hybrid of FBA and PoS. Consensus takes place among quorums and is spread through overlapping nodes (FBA). Users stake their coins within a node and earn rewards on the stake. This serves as economic incentive to operate the node and also as a collateral if node acts maliciously [101].

c: Proof of Burn (PoBurn)

In PoBurn, a node ''burns'' some tokens by sending them to an irretrievable but verifiable address in order to gain mining privilege on the system or generate coins on another system [102]. The miners may be required to burn the native token or some other cryptocurrency, like bitcoin. PoBurn can be used as a migration ''tool'' or bootstrapping a new coin [103].

d: Proof of Disintegration (PoD

PoD is an extension of PoBurn where the coins are not burnt by sending them to an irretrievable & verifiable address, but fully destroys the coins by disintegrating the coin, reducing the circulating and total supply of the coin. PoD is performed on special nodes called ''fundamental nodes'', which yield more staking reward as compared to the normal nodes [104].

e: Proof of History (PoH)

PoH uses the collision resistance property of hashing functions to create a high frequency variable delay function (*VDF*) that can be used to prove that a transaction happened sometime before or after the event. A leader node is randomly chosen from the network to provide a PoH sequence, providing reliable global passage of time. The leader orders and signs the transactions and broadcast them for verifier nodes using the current state of the *VDF*. Verifiers execute the same transactions on their copies of the state and publish their signatures of state as confirmation. This serves as votes in consensus. The hash is obtained on a single core by feeding a random seed and incrementally hashing all hash outputs from previous events and transactions on previous blocks [105].

f: Proof of Process (PoProcess)

PoProcess is based on the idea that every process can be proved by combining the *what* (message digest), *who* (digital signature), *when* (trusted timestamp) and *where* (hashchain) stages of a process into a single proof called link hash. Proof of one process can be included in into another process as a step, forming nested proof of processes [106].

g: Proof of Time (PoTime)

PoTime is a decentralised, off-chain solution for Ethereum to allow scheduled transactions. It is comprised of individual nodes of the decentralized execution network behind the Ethereum Alarm Clock [107] The *timenode's* responsibility is to execute a scheduled transaction and collect reward in return. In order to avoid collision, a *timenode* can claim a transaction by staking small amount of eth. The deposit is lost if the *timenode* goes offline at the time of transaction execution [108].

h: Raft - Quoru

Quorum [97] also uses Raft based consensus protocol. It works on a state replication model, all transactions are replicated across all participating nodes while maintain the sequence of the transactions, regardless of crashes [109].

i: Siev

Sieve was proposed by IBM Research and has been implemented as a part of Hyperledger Fabric. Sieve treats the blockchain as a black box and executes the processes related to non-deterministic operations such as smart contracts and then compare the results. Small number of processes are filtered out if they are detected to create divergence. The whole operation is ''sieved out'' of the sequence if divergence is found among too many processes [110].

j: Tangle

Tangle is a DAG consensus protocol used by IOTA [111]. Each transaction forms a *vertex*, known as *site*. Every new transaction has to approve two previous transactions. This approval represents the *edge* of the graph. Theoretically, a tangle can scale to infinite number of transactions per second. To prevent malicious nodes from spamming the network, each new transaction has to perform a lightweight PoW at the approval stage. Tangle is effective against quantum computer attacks as well [111].

k: Hashgrap

Hashgraph is a proprietary consensus protocol developed by SwirldsTM [112]. Each node maintains its own DAG (Hashgraph) and information is shared in a gossip manner, similar to the blockchains. Vertices or transaction data is called *events* and each *event* carries a creation timestamp which is used in the final ordering. *Events* are hashed along with their history, so each event confirms the entire gossip history. Nodes constantly share the events unknown to their peers, in topological order. The receiving node adds the previously unknown valid events to their graph and at the end of the sync, the receiving node creates and signs a new event that includes any transactions the receiving node intends to submit.

Every event is given two properties, an *id* that puts the node's events in incremental order and a binary value "*witness*", set to true if an event is first created by a particular node in the round. An event is declared "*famous*" if it is a witness and was received by several nodes quickly after the creation. The protocol guarantees that all events will be eventually declared "*famous*" or "*non-famous*" if 2/3 of nodes continue to gossip forever [113]. The network is randomly divided into shards and shards trust each other. Consensus occurs within the shard but shards honour requests from other shards as long as the requesting shard can prove it. Each node votes on the arrival time of a transaction and the median time of all timestamps is used for ordering the transaction.

l: Block-lattic

In Block-lattice, every account gets their own blockchain (*account-chain*) that only they can write to, and everyone holds a copy of all of the chains [114]. The account owner can update its own account-chain asynchronous to the block-lattice. Every transaction is broken down into a send block on the sender's chain and a receive block on the receiving party's chain. There are no overheads in non-conflict transactions and conflicts are resolved via balance-weighted voting. The weight of a node's votew_i is the sum of the balances of all accounts that have named it as its representative. The node keeps a cumulative tally for 4 voting periods totaling up to 1 minute for all incoming votes from M representatives and confirms the winning block. The most popular block b^* will have the majority of the votes and will be retained in the node's ledger.

$$b^* = arg_{b_j} \max v(b_j)$$

Nano has recorded 10,000 transactions per second on a reference implementation of block-lattice [114].

m: SPECTR

SPECTRE (Serialization of Proof-of-work Events) is a proposed improvement on bitcoin's blockchain. It generalizes the blockchain into a DAG (BlockDAG), achieving high scalability. It allows miners to mine block concurrently and with high frequency. SPECTRE requires the miners to embed a list of hashes of all the leaf blocks in the header of newly mined block [115]. It does not produce a linear ordering of the blocks but every block agrees on the pairwise ordering of any two previous blocks. SPECTRE is more suitable for payment networks where totality is more important than the ordering. It is not suitable for execution of smart contracts where ordering is important for computational reasons.

n: PHANTO

PHANTOM is a successor of SPECTRE which produces similar BlockDAG structure but provides a total ordering of the blocks and transactions. This total ordering property makes PHANTOM suitable for smart contracts but it is not as scalable as SPECTRE [33].

IV. USE OF DLTS AND SUITABILITY OF CONSENSUS PROTOCOLS FOR PUBLIC SECTOR

Public sector has shown significant interest in DLTs. Several pilots, case studies and real world applications have been developed utilizing variant forms of DLTs. Estonian government is by far a leader in adopting blockchain in public services. Estonia chose a private permissioned blockchain model for e-services, such as prescriptions, court system, banking, business and land register [119]. According to the Director of Future Borders of UK, HMRC (Her Majesty Revenue and Customs) has "built a proof of concept based on blockchain that demonstrates that you can actually get all of the 28 organizations that act at the border to coordinate all of their risk and intervention" [120].

Similarly, HM Land Registry also recently commissioned a private blockchain (R3 Corda) based pilot project to speed up the conveyancing process and at the same time make it fraudulent proof. Canadian government is using a private-permissioned approach to publish grants information on Ethereum blockchain for public disclosure [121]. "Layer 2" and "side-chain" solutions are also a potential candidate for public sector. Brazil plans to use Ethereum to collect petition data from its citizens. The citizens signs a petition through a mobile app and the root hash of all signatures is published to the blockchain [122].

A pattern in all of the above public sector centric examples is that only a set of privileged nodes (writers) have the write access and all projects are using a private permissioned network.

Birch *et al.* [123] argue that private DLTs should be preferred when a set of privileged group members are responsible for maintaining the integrity of the ledger. Wüstl & Gervais also conclude that permissioned DLTs should be considered when all writers are known [124]. WalPort [116] argues that permissioned DLTs are more suitable for Governments because they allow the owners to enforce rules and limit the usage of the system. The ability of granular control in permissioned DLTs make them more suitable for public sector [117]. This also support the argument in their framework for DLT evaluation. A recent report on DLTs by Campbell *et al.* [118] for Scottish governments also pins down that permissioned DLTs are better suitable for public services as only authorized government actors should access the DLT and have the authority to make changes to it.

Therefore, we believe that the consensus protocols that are designed for public DLTs are automatically less favorable for usage in public services sector (for the reasons highlighted by [116], [117], and [118]). We believe that authority-based consensus protocols with known participants are more suitable for public sector.

We analyzed all protocols discussed in section III regarding their suitability for public and private instances of DLTs, write permissions, overall efficiency of the network, the requirement of built in incentive to compensate the participants, control of an organization on the network events

TABLE 2. Comparison of the reviewed consensus protocols and their suitability for public sector based on the government's preferences. we believe that governments like to work with known and trusted identities and need control.

TABLE 2. *(Continued.)* Comparison of the reviewed consensus protocols and their suitability for public sector based on the government's preferences. we believe that governments like to work with known and trusted identities and need control.

Protocol	Suitable for Private DLTs	Suitable for Public DLTs	Open Write access	Efficiency/High TPS	Incentive	Control on Network	Validators known	Suitability
Delegated BFT (DBFT)	✓	✓		✓	✓	✓	✓	✓
Delayed Proof of Work (DPoW)	✓	✓			✓	✓	✓	✓
Honey Badger Byzantine Fault Tolerance BFT (HB-BFT)	✓	✓		✓	✓	✓	✓	✓
HyderaChain (HC)	✓			✓		✓	✓	✓
Istanbul Byzantine Fault Tolerance BFT (BFT)	✓			✓		✓	✓	✓
MultiChain	✓			✓		✓	✓	✓
Practical BFT (PBFT)	✓	✓		✓		✓	✓	✓
Proof of Authority (PoAuth)	✓	✓		✓		✓	✓	✓
Proof of Elapsed Time (PoET)	✓	✓			✓	✓	✓	✓
Proof of Proof (PoP)	✓	✓		✓	✓	✓	✓	✓
Proof of Personhood (PoPHood)	✓	✓		✓		✓	✓	✓
Proof of Process (PoProcess)	✓	✓				✓	✓	✓
Proof of Reputation (PoRep)	✓	✓		✓	✓	✓	✓	✓
Proof of Stake Boo (PoS-Boo)		✓		✓	✓	✓	✓	✓
Proof of Value (PoVal)	✓	✓		✓	✓	✓	✓	✓
Raft-Quorum (Raft – Q)	✓			✓		✓	✓	✓
Scalable BFT (S-BFT)	✓			✓	✓	✓	✓	✓
Sumeragi	✓			✓		✓	✓	✓
Tendermint	✓			✓		✓	✓	✓
Threshold Relay (T-Relay)	✓	✓		✓		✓	✓	✓
Federated Byzantine Agreement (FBA)	✓	✓		✓			✓	
Limited Confidence Proof of Activity (LCPoA)	✓	✓				✓	✓	
Proof of Signature (PoSign)[2]	✓	✓		✓	✓	✓	✓	
Proof of Time (PoT)		✓			✓	✓	✓	
Ripple Consensus Protocol (RPC)		✓		✓		✓	✓	
Stellar Consensus Protocol (SCP)		✓		✓		✓	✓	
Sieve	✓					✓	✓	
Transaction as Proof of Stake (TPoS)	✓			✓	✓		✓	
BFT-RAFT	✓			✓		✓	✓	✓
BFT-SMART	✓			✓		✓		
Block-lattice		✓	✓	✓	✓			
Delegated Proof of Stake (DPoS)		✓			✓			
Fair Proof of Stake (FPoS)		✓			✓			
Hashgraph	✓	✓	✓	✓				
Hybrid Proof of Work (HPoW)		✓	✓		✓			
Interactive Proof of Stake (PoS)		✓			✓			
Leased Proof of Stake (LPoS)		✓		✓	✓			
Modified Federated Byzantine Agreement (mFBA)	✓	✓		✓	✓			
Magi's Proof of Stake (mPoS)		✓	✓		✓			
Magi's Proof of Work (mPoW)		✓	✓		✓			
Ouroboros	✓	✓	✓	✓	✓			
PHANTOM		✓	✓	✓	✓			
Proof of Activity (PoAct)	✓	✓	✓	✓	✓			
Proof of Approval (PoApr)	✓	✓	✓	✓	✓			
Proof of Burn (PoBurn)		✓	✓		✓			
Proof of Believability (PoBel)	✓	✓	✓	✓	✓			
Proof of Capacity (PoC)		✓	✓		✓			
Proof of Disintegration (PoD)		✓	✓	✓	✓			
Proof of Devotion (PoDev)	✓	✓	✓	✓	✓			
Proof of History (PoH)	✓	✓	✓	✓	✓			
Proof of Importance (PoI)	✓	✓	✓	✓	✓			
Proof of Research (PoRes)		✓	✓		✓			
Proof of Retrievability (PoRetrievability)		✓	✓		✓			
Proof of Stake (PoS)		✓	✓	✓	✓			
Proof of Stake Casper (PoS-Casper)		✓	✓	✓	✓			
Proof of Space Time (PoSpaceTime)		✓	✓		✓			
Proof of Stake Time (PoStakeTime)		✓	✓		✓			

[2] PoSign technically meets the suitability criteria for public sector but it is a proprietary solution, hence we have not considered it as a suitable candidate for public sector.

TABLE 2. ***(Continued.)*** **Comparison of the reviewed consensus protocols and their suitability for public sector based on the government's preferences. we believe that governments like to work with known and trusted identities and need control.**

Proof of Repl (PoRepl)		✓	✓		✓		
Proof of Stake Velocity (PoS-Velocity)		✓	✓	✓	✓		
Proof of Useful Work (PoUW)		✓	✓		✓		
Proof of Work (PoW)		✓	✓		✓		
Proof of Work Time (PoWT)		✓	✓		✓		
Proof of Exercise (PoX)		✓	✓		✓		
SPECTRE		✓	✓	✓	✓		
Tangle		✓	✓	✓			
Transaction as Proof of Stake (TaPoS)		✓	✓	✓			
Trustless Proof of Stake (TPoS)	✓			✓	✓		✓

and knowledge about the participants taking part in the consensus/maintaining the ledger. Public sector can also benefit from the computational power of public DLTs by frequently committing the "backups" to public DLTs. In Table 2, we have highlighted 21 (out of 66 reviewed) consensus protocols suitable for public sector based on the above discussion.

V. CONCLUSION

We have outlined and mapped 66 consensus protocols for private and public DLTs. We believe that no single consensus protocol is a perfect fit for all business needs. One must seriously consider their business needs and deployment environment before choosing the DLT model and consensus protocol involved. The consensus in hostile and untrusted public environment has to be complex and must include incentives and severe penalties for the participant nodes to ensure integrity of the network and to prevent the network from fraudulent nodes. Therefore, security in public DLTs is achieved at the cost of speed and scalability. Conversely, in a private setting with trusted participating nodes, the consensus protocols can be simple and also do not require a reward mechanism as the participating bodies have business interests to protect and secure the network, therefore can focus more on speed and scalability. We have highlighted some consensus protocols suitable for public sector based on the argument that public sector prefers control and authorities on consensus building process, therefore private DLTs are preferable over public DLTs. Paradoxically this contradicts the fundamental decentralization ethos of DLT and vision of Open Government/Data.

As the distinction between digital and physical world is diminishing in an unprecedented rate, data, particularly personal behavioral data is becoming a high valued commodity both for governments and corporations. Therefore, there is a growing call for rights of the citizen to privacy and ownership of personal data. Along with regulations like EU General Data Protection Regulation (GDPR), cryptographic capability of DLT can not only enable "disclosure without exposure" [118] but also can pave the path to "self-sovereign identity"[125].

REFERENCES

[1] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, Jul. 1982.

[2] K. P. Birman, "Replication and fault-tolerance in the ISIS system," *ACM SIGOPS Oper. Syst. Rev.*, vol. 19, no. 5, pp. 79–86, 1985.

[3] L. Lamport, "Paxos made simple," *ACM SIGACT News*, vol. 32, no. 4, pp. 18–25, 2001.

[4] D. Ongaro and J. K. Ousterhout, "In search of an understandable consensus algorithm," in *Proc. USENIX Annu. Tech. Conf.*, 2014, pp. 305–319.

[5] B. M. Oki and B. H. Liskov, "Viewstamped replication: A new primary copy method to support highly-available distributed systems," in *Proc. 7th Annu. ACM Symp. Princ. Distrib. Comput.*, 1988, pp. 8–17.

[6] M. Burrows, "The Chubby lock service for loosely-coupled distributed systems," in *Proc. 7th Symp. Oper. Syst. Design Implement.*, 2006, pp. 335–350.

[7] P. Hunt, M. Konar, F. P. Junqueira, and B. Reed, "ZooKeeper: Wait-free coordination for internet-scale systems," in *Proc. USENIX Annu. Tech. Conf.*, 2010, vol. 8, no. 9, pp. 1–14.

[8] M. Castro and B. Liskov, "Practical byzantine fault tolerance and proactive recovery," *ACM Trans. Comput. Syst.*, vol. 20, no. 4, pp. 398–461, 2002.

[9] M. J. Fischer, N. A. Lynch, and M. S. Paterson, "Impossibility of distributed consensus with one faulty process," *J. ACM*, vol. 32, no. 2, pp. 374–382, 1985.

[10] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: http://Www.Bitcoin.Org

[11] A. Back, "Hashcash—A denial of service counter-measure," Tech. Rep., 2002.

[12] L. M. Bach, B. Mihaljevic, and M. Zagar, "Comparative analysis of blockchain consensus algorithms," *Proc. 41st Int. Conv. Inf. Commun. Technol. Electron. Microelectron. (MIPRO)*, May 2018, pp. 1545–1550.

[13] C. Cachin and M. Vukolić, "Blockchain consensus protocols in the wild," no. 15, pp. 1–16, 2017.

[14] G. T. Nguyen and K. Kim, "A survey about consensus algorithms used in Blockchain," *J. Inf. Process. Syst.*, vol. 14, no. 1, pp. 101–128, 2018.

[15] M. Jun, "Blockchain government—A next form of infrastructure for the twenty-first century," *J. Open Innov. Technol. Market Complex.*, vol. 4, no. 1, p. 7, 2018.

[16] S. Ølnes, J. Ubacht, and M. Janssen, "Blockchain in government: Benefits and implications of distributed ledger technology for information sharing," *Government Inf. Quart.*, vol. 34, no. 3, pp. 355–364, 2017.

[17] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang, "Untangling blockchain: A data processing view of blockchain systems," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1366–1385, Jul. 2018.

[18] X. Xu *et al.*, "A taxonomy of blockchain-based systems for architecture design," in *Proc. IEEE Int. Conf. Softw. Archit.*, Apr. 2017, pp. 243–252.

[19] S. Bano *et al.* (2017). "Consensus in the age of blockchains." [Online]. Available: https://arxiv.org/abs/1711.03936

[20] A. Tapscott and D. Tapscott, "How blockchain is changing finance," *Harvard Bus. Rev.*, vol. 1, no. 9, pp. 1–5, 2017.

[21] J. Yarbrough and A. K. Mirkovic, "Blockchain pilot program. Final report," Deputy Rec. Deeds, Commun./IT, Cook County, IL, USA, Tech. Rep., 2017. [Online]. Available: http://cookrecorder.com/wp-content/uploads/2016/11/Final-Report-CCRD-Blockchain-Pilot-Program-for-web.pdf

[22] R. Chandran, "Indian states look to digitize land deals with blockchain," Thomson Reuters Corp., Eagan, MN, USA, Tech. Rep., Aug. 2017.

[23] N. Kshetri, "Blockchain's roles in strengthening cybersecurity and protecting privacy," *Telecommun. Policy*, vol. 41, no. 10, pp. 1027–1038, 2017.

[24] S. A. Abeyratne and R. P. Monfared, ‘‘Blockchain ready manufacturing supply chain using distributed ledger,’’ Tech. Rep., 2016.

[25] F. Tian, ‘‘An agri-food supply chain traceability system for China based on RFID & blockchain technology,’’ in *Proc. 13th Int. Conf. Service Syst. Service Manage. (ICSSSM)*, Jun. 2016, pp. 1–6.

[26] M. B. M. Micheal, ‘‘Chain reaction: How blockchain technology might transform wholesale insurance,’’ *Nursing Times*, vol. 94, no. 37, pp. 36–37, 2015.

[27] T.-T. Kuo, H.-E. Kim, and L. Ohno-Machado, ‘‘Blockchain distributed ledger technologies for biomedical and health care applications,’’ *J. Amer. Med. Inform. Assoc.*, vol. 24, no. 6, pp. 1211–1220, 2017.

[28] *Illinois Opens Blockchain Development Partnership With Hashed Health*, Illinois Blockchain Initiative, Aug. 2017. Accessed: Aug. 14, 2018. [Online]. Available: https://illinoisblockchain.tech/illinois-opens-blockchain-development-partnership-with-hashed-health-fe3891e500bb

[29] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, ‘‘On blockchain and its integration with IoT. Challenges and opportunities,’’ *Futur. Gener. Comput. Syst.*, vol. 88, pp. 173–190, Nov. 2018.

[30] T. M. Fernández-Caramés and P. Fraga-Lamas, ‘‘A review on the use of blockchain for the Internet of Things,’’ *IEEE Access*, vol. 6, pp. 32979–33001, 2018.

[31] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, ‘‘An overview of blockchain technology: Architecture, consensus, and future trends,’’ in *Proc. IEEE 6th Int. Congr. Big Data, BigData Congr.*, Jun. 2017, pp. 557–564.

[32] P. Ferraro, C. King, and R. Shorten. (2018). ‘‘Distributed ledger technology, cyber-physical systems, and social compliance.’’ [Online]. Available: https://arxiv.org/abs/1807.00649

[33] Y. Sompolinsky and A. Zohar, ‘‘PHANTOM, GHOSTDAG: Two scalable BlockDAG protocols,’’ Tech. Rep.

[34] S. Haber and W. S. Stornetta, ‘‘How to time-stamp a digital document,’’ *J. Cryptol.*, vol. 3, no. 2, pp. 99–111, 1991.

[35] N. Hutton, J. Maloberti, S. Nickel, T. F. Rønnow, J. J. Ward, and M. Weeks, ‘‘Design of a scalable distributed ledger,’’ Tech. Rep., 2018, pp. 1–30.

[36] Y. Lewenberg, Y. Sompolinsky, and A. Zohar, ‘‘Inclusive block chain protocols,’’ in *Proc. Int. Conf. Financial Cryptogr. Data Secur.*, 2015, pp. 528–547.

[37] A. Baliga, ‘‘Understanding blockchain consensus models,’’ Persistent Syst. Ltd., Apr. 2017.

[38] F. Tschorsch, ‘‘Bitcoin and beyond a technical survey on.pdf,’’ Tech. Rep.

[39] C. Dwork and M. Naor, ‘‘Pricing via Processing or Combatting Junk Mail,’’ in *Proc. Annu. Int. Adv. Cryptol. (CRYPTO)*, 1992, pp. 139–147.

[40] *Komodo: An Advanced Blockchain Technology, Focused on Freedom*.

[41] M. Sanchez, ‘‘PoP-white-paper,’’ Tech. Rep., pp. 1–22.

[42] W. B, ‘‘Lynx Technical White Paper 1.0,’’ Tech. Rep., pp. 1–28.

[43] L. Chen, L. Xu, N. Shah, Z. Gao, Y. Lu, and W. Shi, ‘‘On security analysis of proof-of-elapsed-time (PoET),’’ in *Proc. Int. Symp. Stabilization, Saf., Secur. Distrib. Syst.*, Oct. 2017, pp. 282–297.

[44] M. Milutinovic, W. He, H. Wu, and M. Kanwal, ‘‘Proof of luck: An efficient blockchain consensus protocol,’’ in *Proc. SysTEX*, 2017, pp. 2–7.

[45] A. Shoker, ‘‘Sustainable blockchain through proof of exercise,’’ in *Proc. IEEE 16th Int. Symp. Netw. Comput. Appl. (NCA)*, Jan. 2017, pp. 1–9.

[46] M. Ball, A. Rosen, M. Sabin, and P. N. Vasudevan, ‘‘Proofs of useful work,’’ *IACR Cryptol. ePrint Arch.*, vol. 2017, p. 203, Feb. 2017.

[47] *Gridcoin White Paper the Computation Power of a Blockchain Driving Science & Data Analysis*, Gridcoin, pp. 1–12.

[48] D. P. Anderson, ‘‘BOINC: A system for public-resource computing and storage,’’ in *Proc. 5th IEEE/ACM Int. Workshop Grid Comput.*, Nov. 2004, pp. 4–10.

[49] M. Rosenfeld. (2011). ‘‘Analysis of bitcoin pooled mining reward systems.’’ [Online]. Available: https://arxiv.org/abs/1112.4980

[50] *Proof-of-Work-Time-VeriCoin &amp; Verium Wiki*. Accessed: Oct. 17, 2018. [Online]. Available: https://wiki.vericoin.info/index.php?title=Proof-of-Work-Time

[51] J. Lao. (2014). ‘‘A network-dependent rewarding system: Proof-of-mining.’’ [Online]. Available: https://arxiv.org/abs/1409.7948

[52] A. Kiayias *et al.*, ‘‘PPCoin: Peer-to-peer crypto-currency with proof-of-stake,’’ in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, vol. 1919, Jan. 2017, pp. 1–27.

[53] Bitshares. (2017). *Delegated Proof-of-Stake Consensus*. Accessed: Sep. 18, 2018. [Online]. Available: https://bitshares.org/technology/delegated-proof-of-stake-consensus/

[54] A. Begicheva and A. Kofman, ‘‘Fair proof of stake,’’ Tech. Rep., 2018, pp. 1–13.

[55] A. Chepurnoy, ‘‘Interactive Proof-of-stake,’’ Tech. Rep., 2016.

[56] *Shield White Paper v1.0.2*, The Shield Team.

[57] D. Pike, P. Nosker, D. Boehm, D. Grisham, S. Woods, and J. Marston, ‘‘PoST white paper,’’

[58] L. Ren, ‘‘Proof of stake velocity: Building the social currency of the digital age,’’ Self-Published White Paper, 2014, pp. 1–13.

[59] V. Buterin and V. Griffith. (2017). ‘‘Casper the friendly finality gadget.’’ [Online]. Available: https://arxiv.org/abs/1710.09437

[60] The Coin MAGI Project. *MAGI|Coin MAGI*. Accessed: Sep. 28, 2018. [Online]. Available: https://www.m-core.org/resources/mining.html#mpos-mining

[61] D. Larimer, ‘‘Transactions as proof-of-stake!’’ Tech. Rep., 2013, pp. 1–9.

[62] *Stake Net Whitepaper*, Jul. 2018, p. 55. Accessed: Jul. 14, 2018. [Online]. Available: https://stakenet.io/Whitepaper_Stakenet_V3.0_EN.pdf

[63] A. Kiayias, A. Russell, B. David, and R. Oliynykov, ‘‘Ouroboros: A provably secure proof-of-stake blockchain protocol,’’ in *Advances in Cryptology—CRYPTO* (Lecture Notes in Computer Science), vol. 10401. Cham, Switzerland: Springer, 2017, pp. 357–388.

[64] S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, ‘‘PBFT vs proof-of-authority: Applying the CAP theorem to permissioned blockchain,’’ in *Proc. CEUR Workshop*, 2018, pp. 1–11.

[65] *GoChain: Blockchain at Scale*, GoChain, 2018, p. 5.

[66] J. K. Liu, V. K. Wei, and D. S. Wong, ‘‘Linkable spontaneous anonymous group signature for ad hoc groups,’’ in *Proc. Australas. Conf. Inf. Secur. Privacy*, 2004, pp. 325–335.

[67] E. Syta *et al.*, ‘‘Keeping authorities ‘honest or bust’ with decentralized witness cosigning,’’ in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2016, pp. 526–545.

[68] E. Syta *et al.*, ‘‘Scalable bias-resistant distributed randomness,’’ in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 444–460.

[69] M. Borge, E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, and B. Ford, ‘‘Proof-of-personhood: Redemocratizing permissionless cryptocurrencies,’’ in *Proc. 2nd IEEE Eur. Symp. Secur. Privacy Workshop (EuroS PW)*, Apr. 2017, pp. 23–26.

[70] G. Greenspan, ‘‘Multichain private blockchain,’’ White Paper, 2013, pp. 1–17.

[71] *Xtrabytes Non-Technical*. Accessed: Jul. 14, 2018. [Online]. Available: https://xtrabytes.global/build/files/whitepaper.pdf

[72] S. Takahashi, ‘‘Proof-of-approval: A distributed consensus protocol for blockchains,’’ Tech. Rep., 2018, pp. 1–21.

[73] *Internet of Services: The Next-Generation, Secure, Highly Scalable Ecosystem for Online Services*, 2017, pp. 1–23. Accessed: Jul. 18, 2018. [Online]. Available: https://iost.io/iost-whitepaper/

[74] NEM Foundation. (Feb. 23, 2018). *NEM: Technical Reference*. Accessed: Jul. 14, 2018. [Online]. Available: https://nem.io/wpcontent/themes/nem/files/NEM_techRef.pdf

[75] A. N. Langville and C. D. Meyer, *Google's PageRank and Beyond: The Science of Search Engine Rankings*. Princeton, NJ, USA: Princeton Univ. Press, 2011.

[76] *Nebulas Technical White Paper*, 2018, pp. 1–60. Accessed: Jul. 16, 2018. [Online]. Available: https://nebulas.io/docs/NebulasTechnicalWhitepaper.pdf

[77] *Backfeed: Decentralized Value Distribution System for Blockchain-Based Applications*.

[78] A. Bhatia, R. Giometti, and A. Nicolas, ‘‘Decentralized social news platform,’’ Tech. Rep., Mar. 2018, pp. 1–52.

[79] *AI Crypto: A Blockchain for Decentralized Economy*. Accessed: Jul. 15, 2018. [Online]. Available: https://www.aicrypto.ai/AIC_WhitePaper_Kor.pdf

[80] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, ‘‘Proof of activity: Extending bitcoin's proof of work via proof of stake,’’ *IACR Cryptol. ePrint Arch.*, vol. 452, no. 3, pp. 1–19, 2014.

[81] A. Nedobylsky. *LCPoA–Universal as PoW, Economical as PoS–IZZZIO-Medium*. Accessed: Sep. 28, 2018. [Online]. Available: https://medium.com/@izzzio/lcpoa-universal-as-pow-economical-as-pos-c26f6ba90017

[82] S. Gauld, F. Von Ancoina, and R. Stadler, ‘‘The burst dymaxion an arbitrary scalable, energy efficient and anonymous transaction network based on colored tangles,’’ in *Proc. CryptoGuru PoC SIG*, 2017.

[83] A. Miller, A. Juels, E. Shi, B. Parno, and J. Katz, "Permacoin: Repurposing bitcoin work for data preservation," in *Proc. IEEE Symp. Secur. Privacy*, May 2014, pp. 475–490.

[84] C. Reitwiessner, "zkSNARKs in a nutshell," *Ethereum Blog*, vol. 6, 2016.

[85] J. Benet and N. Greco, "Filecoin: A decentralized storage network," Tech. Rep., 2018, pp. 1–36.

[86] J. Benet, D. Dalrymple, and N. Greco, "Proof of replication," Tech. Rep., 2017, pp. 1–10.

[87] C. Dwork, N. Lynch, and L. Stockmeyer, "Consensus in the presence of partial synchrony," *J. ACM*, vol. 35, no. 2, pp. 288–323, 1988.

[88] J. Kwon, "Tendermint: Consensus without mining," Tech. Rep., 2014, pp. 1–10.

[89] S. Duan, H. Meling, S. Peisert, and H. Zhang, "BChain: Byzantine replication with high throughput and embedded reconfiguration," in *Proc. Int. Conf. Princ. Distrib. Syst.*, 2014, pp. 91–106.

[90] *Hyperledger Architecture*, Hyperledger Archit. Work. Group, 2017, vol. 1, p. 15.

[91] T. Hanke, M. Movahedi, and D. Williams. (2018). "DFINITY technology overview series, consensus system." [Online]. Available: https://arxiv.org/abs/1805.04548

[92] A. Bessani, J. Sousa, and E. E. P. Alchieri, "State machine replication for the masses with BFT-SMART," in *Proc. 44th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Nov. 2014, pp. 355–362.

[93] C. Copeland and H. Zhong, "Tangaroa: A byzantine fault tolerant raft," Tech. Rep., 2016.

[94] (2017). *NEO White Paper*. Accessed: Sep. 19, 2018. [Online]. Available: http://Docs.Neo.Org/En-Us/Index.Html and http://docs.neo.org/en-us/whitepaper.html

[95] (2015). *HyderaChain Consensus Explained*. Accessed: Sep. 22, 2018. [Online]. Available: https://github.com/HydraChain/hydrachain/blob/develop/hc_consensus_explained.md

[96] A. Miller, Y. Xia, K. Croman, E. Shi, and D. Song, "The honey badger of BFT protocols," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, vol. 3, 2016, pp. 31–42.

[97] *Quorum GitHub*, Jpmorganchase, New York, NY, USA. Accessed: Nov. 28, 2018. [Online]. Available: https://github.com/jpmorganchase/quorum

[98] J. Behl and T. Distler, "Scalable BFT for multi-cores: Actor-based decomposition and consensus-oriented parallelization," in *Proc. 10th Workshop Hot Topics Syst. Dependability (HotDep)*, 2014.

[99] D. Mazières, "The stellar consensus protocol: A federated model for Internet-level consensus," Tech. Rep., 2015, pp. 1–45.

[100] *Ripple Solution Overview*, Ripple, 2017.

[101] H. Park, C. Park, Y. Choi, and J. H. Choi, "The BOScoin white paper," *BOScoin*, vol. 1, no. 1, pp. 1–17, 2016. Accessed: Nov. 30, 2018. [Online]. Available: http://boscoin.net/BOScoinWhitePaperv20170121.pdf

[102] *Slimcoin: A Peer-to-Peer Crypto-Currency With Proof-of-Burn*, P4Titan, 2014.

[103] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Commun. Surveys Tuts.*, vol. 18, no. , pp. 2084–2123, 3rd Quart., 2016.

[104] *B3 Coin—Proof of Disintegration and Fundamental Node*. Accessed: Oct. 11, 2018. [Online]. Available: https://b3coin.io/#info

[105] A. Yakovenko. *Solana: A New Architecture for a High Performance Blockchain*. Accessed: Nov. 30, 2018. [Online]. Available: https://solana.com/solana-whitepaper.pdf

[106] *Proof of Process*, Stratumn, Paris, France, 2016.

[107] *Ethereum Alarm Clock*. Accessed: Oct. 17, 2018. [Online]. Available: https://www.ethereum-alarm-clock.com/

[108] *Temporal Innovation on the Blockchain*, Chronologic, 2016.

[109] *Raft Documentation Github*, Jpmorganchase. Accessed: Nov. 29, 2018. [Online]. Available: https://github.com/jpmorganchase/quorum/wiki

[110] C. Cachin, S. Schubert, and M. Vukolić. (2016). "Non-determinism in byzantine fault-tolerant replication." [Online]. Available: https://arxiv.org/abs/1603.07351

[111] S. Popov, "The tangle," IOTA, White Paper, 2017, pp. 1–28.

[112] *Home—Swirlds*. Accessed: Sep. 20, 2018. [Online]. Available: https://www.swirlds.com/

[113] L. Baird, "The swirlds hashgraph consensus algorithm: Fair, fast, byzantine fault tolerance," Tech. Rep. SWIRLDS-TR-2016-01, 2016.

[114] C. Lemahieu, "Nano: A feeless distributed cryptocurrency network," *Nano*, pp. 1–8, 2014. Accessed: Nov. 25, 2018. [Online]. Available: https://nano.org/en/whitepaper

[115] Y. Sompolinsky, Y. Lewenberg, and A. Zohar, "SPECTRE: Serialization of proof-of-work events: confirming transactions via recursive elections," Tech. Rep., 2016.

[116] M. Walport, "Distributed ledger technology: Beyond blockchain," Tech. Rep., 2015, pp. 1–88.

[117] C. Mulligan, J. Z. Scott, S. Warren, and J. Rangaswami, "Blockchain beyond the hype; a practical framework for business leaders," World Economic Forum, White Paper, Apr. 2018.

[118] R. H. C. Rab, T. Gillian, and F. Peter, "Distributed ledger technologies in public services," Tech. Rep., Jun. 2018.

[119] (2017). *Frequently Asked Questions: Estonian Blockchain Technology*. [Online]. Available: https://e-estonia.com/

[120] E. Lis. (2017). *HMRC Builds Blockchain Proof of Concept for UK Border*. Accessed: Nov. 25, 2018. [Online]. Available: https://www.computerweekly.com/news/450426393/HMRC-builds-blockchain-proof-of-concept-for-UK-border

[121] *Catena*. Accessed: Nov. 1, 2018. [Online]. Available: https://explorecatena.com/

[122] *Ethereum Could be the Solution Brazil'S Constitutionally Guaranteed Petitions—Quartz*. Accessed: Nov. 1, 2018. [Online]. Available: https://qz.com/1163660/brazil-may-write-new-laws-based-on-data-stored-on-the-ethereum-blockchain/

[123] D. Birch, R. G. Brown, and S. Parulava, "Towards ambient accountability in financial services: Shared ledgers, translucent transactions and the technological legacy of the great financial crisis," *J. Payments Strateg. Syst.*, vol. 10, no. 2, pp. 118–131, 2016.

[124] K. Wüst and A. Gervais, "Do you need a Blockchain?" *IEEE Spectr.*, in *Proc. Crypto Valley Conf. Blockchain Technol. (CVCBT)*, Jun. 2018, pp. 45–54.

[125] A. Christopher. (2016). *The Path to Self-Sovereign Identity*. Accessed: Nov. 25, 2018. [Online]. Available: http://www.lifewithalacrity.com/2016/04/the-path-to-self-sovereign-identity.html

Authors' photographs and biographies not available at the time of publication.