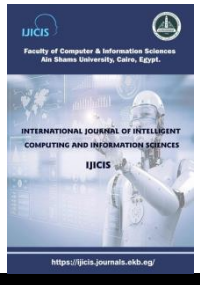
IJICIS, Vol.23, No.4, 1-18
DOI: 10.21608/ijicis.2023.236431.1294

International Journal of Intelligent Computing and Information Sciences

https://ijicis.journals.ekb.eg/



SELECTION CRITERIA FOR INDUSTRIAL DISTRIBUTED LEDGER TECHNOLOGY

Mohamed Ahmed Abo-Soliman*

Computer Systems Department,
Faculty of Computer and Information Science, Ain Shams University
Cairo, Egypt
mohamed.abosoliman@cis.asu.edu.eg

Eman Shaaban

Computer Systems Department,
Faculty of Computer and Information Science, Ain Shams University
Cairo, Egypt
Eman.Shaaban@cis.asu.edu.eg

Mirvat Al-Qutt

Computer Systems Department
Faculty of Computer and Information Science, Ain Shams University
Cairo, Egypt
mmalqutt@cis.asu.edu.eg

Kareem Emara

Computer Systems Department
Faculty of Computer and Information Science, Ain Shams University
Cairo, Egypt
Karim.Emara@cis.asu.edu.eg

Received 2023-09-30; Revised 2023-09-30; Accepted 2023-12-07

Abstract: *The use of distributed ledger technology for industrial IoT devices is increasing recently to ensure network security and data protection. Factories and manufacturing plants inclined lately to deploy both Industrial IoT and DLT applications in order to enable autonomous secure operations. IoT helps in simplifying business processes, improving user's experience and leading to better cost efficiencies, while DLT ensures security, transparency and trust. DLT-based IoT supports secure automation for industrial systems and fosters transformation into the industry 4 age. However, DLT still faces several challenges such as scalability, high cost and security. Moreover, there is no clear understanding about DLT-IoT architecture by a wide range of the industrial community. This work introduces DLT as a major key component of industrial IoT systems that benefits the industry with high level of protection and trust. It also surveys different DLT consensus with regard to industry in order to construct a comparative analysis between the most common algorithms. The study concludes by a selection criteria chart for building integrated DLT-IoT solution suitable for different types of businesses.*

Keywords: *Blockchain, Consensus, Distributed Ledger Technology, Industrial IoT, Security.*

1. Introduction

***Corresponding Author**: Mohamed Ahmed Abo-Soliman

Computer Systems Department, Faculty of Computer and Information Science, Ain Shams University, Cairo, Egypt

Email address: mohamed.abosoliman@cis.asu.edu.eg

Industry 4 encourages to a complete transformation of the manufacturing operations into a fully automated distributed systems. This is applied through some new technologies such as robots, artificial intelligence, big data, and IoT. Industrial IoT (IIoT) is the core of smart manufacturing[1] solutions, where a large number of distributed components are utilized. ICS (Industrial Control Systems) are used to manage the manufacturing processes through DCS (Distributed Control Systems) which include hardware, software and connections. Collaboration between these components promises an autonomous, reliable, high quality and flexible production. However, various security challenges in terms of authentication, access control and data integrity are encountered. Therefore, Distributed Ledger Technologies (DLT) is applied as a security technique for IIoT environments to overcome IIoT security challenges. DLT was initially proposed for worldwide money transfer and a medium for exchanging digital-currency. Yet, its usage has extended nowadays for many other purposes such as providing network security and data integrity in IoT. DLT is the uprising security technique for intercommunication through public medium. It is one of the most reliable verification mechanisms, which allows transparency, immutability, and trust. In this context, this work surveys the most common types of DLT with regard to IoT applications. It also presents a comparative analysis between the most common consensus-algorithms used in DLTs. The main objective of the study is to figure out a practical workflow that support in building IoT-DLT solution that meets all industries. After this introduction, section 2 discusses IIoT and its key-components. Section 3 introduces DLT, while Section 4 deeply surveys DLT common algorithms. Section 5 introduces a decision chart that recommends the main criteria to consider for building industrial DLT-IoT. Finally, the conclusion and future work are discussed in section 6.

2. IIoT

Industrial IoT is a group of smart physical devices connected via network with embedded sensors, software, applications, and other components to collect, provide, and exchange data that support in making complex decisions. The IIoT term may represent M2M (machine-to-machine) communication, cyber-physical systems or sensor networks or any other things that support the manufacturing processes. IIoT integrates IP sensing devices with internet to allow data transmission from things to things or things to humans. This network of things including almost all electronic machines communicates measured data and transfers real world information that support in applying a proper action or taking a suitable decision. Amazon AWS IoT and Google Cloud IoT are known examples of IoT networks. According to IDC[2], there will be 55.7 connected devices in the world by 2025. 75% of which will be connected to IoT platform. There would be 73.1 ZB generated data by those devices.

2.1. IIoT Architecture and Relative Attacks

IIoT comprises three main layers, the application layer, the network layer, and the perception layer. The application layer usually deployed as cloud-based applications such as supply chain, object tracking, monitoring inventory, and quality control. The network layer represents the networking devices and techniques such as routers, wireless access points, gateways and other connecting technologies that connects the perception layer with the application layer. The perception layer includes various sensing devices that collects and measures the industrial conditions. Each layer undergoes relative attacks. A list of attacks that targets each layer are surveyed in [3] while [4] also reviews IoT attacks and provides proposed mitigations for such attacks.

2.2. Industrial Revolution Key Components

The global industry witnessed several revolutions through ages. *"Industry 4"* is the current title of today's industry. The recent generation industry aims at re-engineering the traditional business model by interconnecting different manufacturing facilities through cloud-based IoT architecture[5]. Nowadays, Manufacturers prefer to build a full suite of unmanned autonomous factories and production platforms depending on the huge number of internet-based connected devices, the various implementations of cyber physical systems. IIoT integrates recent technologies such as smart sensors, robots, M2M, big data analytics and AI (Artificial Intelligence) for handling normal operational processes and procedures[6], [7], [8]. The vision of a complete smart factory [9] can be reached through IIoT which allows horizontal data flow between suppliers, customers and involved technology partners, in additions to allowing vertical messages between industrial systems from development to final product[10]. It also facilitates smart factory processes by simulating the physical world by connecting manufacturing devices that can consequently make decentralized decisions. According to [7], there are five key technologies for IIoT; seen in Figure 1: big data, smart robotics, IoT, AI and blockchain.

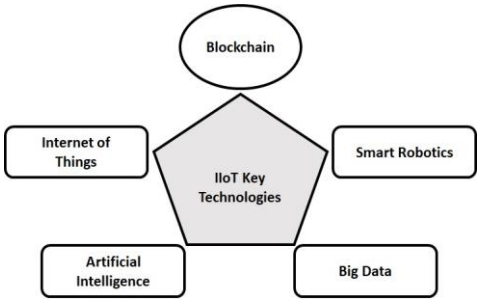


Figure 1: IIoT Key Components

3. DLT

DLT is a decentralized, distributed, shared, and immutable database ledger that stores users' transactions across a P2P network[3]. DLT is known as "The Future of Computing" and is currently used in various fields since it allows transparent and verifiable transactions while preserving privacy. It is considered one of the most reliable verification mechanisms for a group of unreliable parties to perform trusted and verified transactions. The core concept behind DLT is migrating data management from a central authority to distributed managed systems. Actually, this concept is recently appealing to a very large community. For the industry, DLT is a promising platform that helps to accomplish a complete autonomous smart factory. Industrial DAPPs (distributed applications) on top of DLT simplify the process of tracking and storing communicated digital messages. Smart factories can use DLT-based algorithms to record transmitted messages between smart IIoT devices. Blockchain is a common implementation for DLT that stores data in a form of linked list of cryptographically ordered blocks. It is stated by the international data corporation [11] that 20% of deployed IoT will offer blockchain-enabled services by 2027. More than 10% of global GDP will be related to blockchain-enabled systems. In this section, we discuss DLT architecture, advantages, Challenges and recent techniques.

3.1. DLT Background and Traditional Architecture

Although DLT was introduced long time ago in its classical form to enable data fault tolerance, its usage has expanded lately when Satoshi Nakamoto introduced Bitcoin as a new electronic digital currency

eligible for worldwide transactions. Bitcoin cryptocurrency network allows participating persons to transfer money among them easily and globally without relying on a third party such as banks or other financial or governmental entities. Bitcoin supports in transforming the current centrally-managed financial and banking system into a distributed immutable ledger[12]. DLT later flourished and the capital market for bitcoin reached more than 350 billion US Dollars in late 2020. There are more than 50 million active investors who trade in bitcoin and other cryptocurrencies in more than 100 exchange worldwide[13]. DLT is a term, which represents all techniques of recording distributed data records and verify these records by network members. Traditional blockchain is a cryptographically linked list of blocks created by participants. Committed transactions are shared between participants as a public ledger and are verified by a majority consensus of miners. Miner nodes are actively involved in verifying and validating transactions in order to help in defending against modification or compromising. The series of generated linked blocks store a list of completed transaction records. These blocks are chronologically ordered and sequentially linked to each other's. This link is applied by recording the previous block hash at the header of the current block and the current block hash is recorded at the next block header. The first block is called the genesis block since it has no parent block. Each block comprises a header and a body. The block header contains the embedded security metadata that includes a timestamp, parent block hash, Merkle tree root hash, Nonce, Nbits, and the block version[14]. The body of the block includes a list of approved transactions and transaction counter. Figure 2 displays how blocks are sequentially linked.

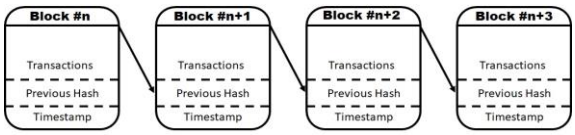


Figure 2: The Linked Blocks in Blockchain

3.2. Recent DLT Techniques

Unlike Nakatomo linear block based distributed ledger, many other DLT were introduced to cope with different fields and resolve previous blockchain problems. They differ in the way of validating data or even in the data structure, data sharing and processing. The state machine approach[15] is a well-known technique that implements fault-tolerance between a group of replicating servers. In a "State Machine Replication", there are multiple server-replicas, which have the same state and same data to ensure the network operation even in case of some faulty nodes. Figure 3 displays a simple replicated state machine. Another technique of data verification is based on DAG (Directed Acyclic Graph)[16]. In DAG-based DLT, there is no leader election nor blocks of data. Rather, transactions are added one by one to the ledger instead of block formation. Figure 4 displays a DAG based data verification.

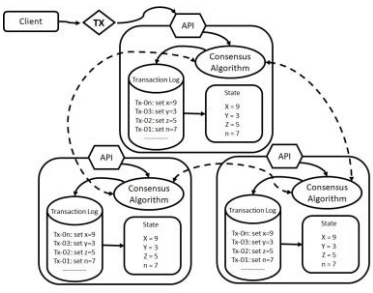


Figure 3: State Machine Replication

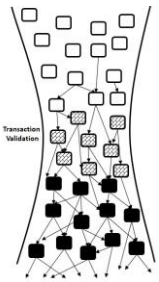


Figure 3: DAG Verification

3.3. DLT Advantages for IIoT

DLT supports IIoT with the following benefits:

- Cost reduction: Installing DLT programs to the participants' devices or networked nodes avoids the cost of central management and DB servers. The network members generate, verify and manage data.
- Enhanced security: DLT natively provides data validation, privacy, transparency and trust.
- High availability: DLT shares the data ledger among all participating nodes. As a result there is no risk of "*single point of failure*"
- DAPPs: implementing distributed applications based on the blockchain smart contract facility provides secure and trusted programs.

3.4. Traditional DLT Issues and Challenges

There are two main types of challenges when deploying DLT for IoT; general DLT challenges and IoT related challenges. The general DLT challenges include:

- Security attacks: DLT subjects to several security compromises such as double spending, Cybil attack, forking and the 51% attack. Security attacks that target Blockchain are studied in, [17]
- Performance issues: Traditional DLT consumes high power and requires high computing resources in order to apply its data validation processes.
- Scalability: the size of the distributed ledger increases proportionally with the network expansion and with the increased activities of its participants. This requires higher computing power and storage with respect to that increase.

The second type of DLT challenges that face DLT-IoT in particular include:

- Record confirmation time: adding a new transaction to a ledger typically takes time for data validation. This may be inadequate for critical industries that require real time transactions.
- IoT constrained devices: tiny industrial sensors of low computing and power cannot afford the required resource-greediness of DLT.

4. Consensus Algorithms

Consensus algorithm is *a* key element at the core of any DLT. It allows a high trust between decentralized forms of data shared among different entities who are untrusted to each other's. To ensure validity of a transaction, message, event or any new data record, a node is selected to approve this data record. The process of node-selection generally called "mining" in legacy blockchains or "Leader-election" in other DLTs. New data blocks including transactions are appended to the chain based on the agreement decided by the consensus algorithms. POW (Proof of Work), POS (Proof of stake), DPOS (Delegated Proof of Stake), proof of elapsed time, PBFT (Practical Byzantine Fault Tolerance), Ripple, Tendermint, and Tangle are common examples of consensus. This section classifies the existing consensus techniques with examples of real-life algorithms and IIoT relative elaboration. In a recent research [18], consensus algorithms are categorized into two main categories: proof-based consensus, and voting-based consensus. In proof-based consensus, the node who performs sufficient proof will get the right to append a new block to the chain and receives the reward. While in voting-based consensus, network members exchange their verification results to allow addition of new data to the ledger. POW and POS are examples of the first type and often used in public blockchains, while Stellar, Tendermint, Tangle and RAFT are examples of voting-based consensus and often preferred in private and consortium blockchains. Voting based

consensus are also divided into two subclasses: BFT (Byzantine Fault Tolerance) and CFT (Crash Fault Tolerance). Figure 5 displays the main categories of consensuses with relative examples.

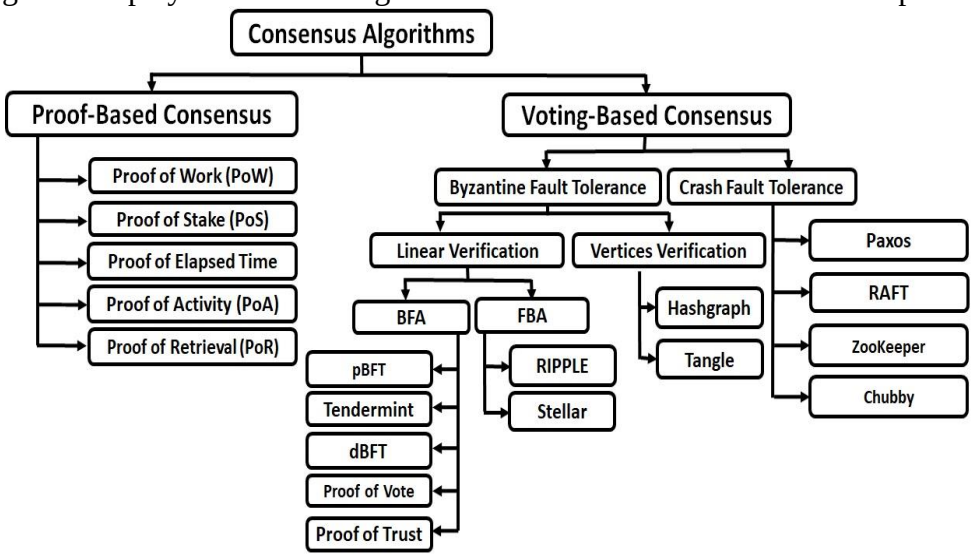


Figure 4: Main Categories of Consensus

4.1. Proof Based Consensus

In proof-based consensus, each participating node endeavors by itself to show that it is qualified enough to win the preconfigured custom competition and that it is accountable to add the next block. In public blockchain, the winning node usually receives a reward after appending the new block in order to encourage more participation and ensures the chain validity. POW consensus is the first consensus model that was introduced in BitCoin. The node must solve a difficulty adjustable puzzle in order to append a new block to the chain. The first node which succeed in solving this puzzle will have the right to append a new block and earn a reward. The effort of solving the puzzle is called mining. Unfortunately, the computational operation exerted in mining a block consumes high power and computing resources and would takes about 10 minutes, which may be inadequate for some domains other than money transactions. In additions, PoW is subjected to security issues such as forking, Sybil and 51% attacks[19]. Several variants of proof-based algorithms were introduced to overcome PoW performance and security issues. PoS [20], DPoS [21], PoA [22], PoB [23] and PoET [24] are common proof based consensus algorithms.

4.2. Voting Based Consensus

In voting-based techniques, nodes that perform verification are elected by some or all other nodes based on a voting mechanism. The list of verifying nodes is dynamically changing where nodes can be added or removed based on the election mechanism. New transactions are added to the ledger only after receiving information about the same proposed block from a configured minimum number of participants. Voting-based consensus is designed to react to system failures. There are two main types of failures: crash failures and Byzantine generals' problem failure. The crash failure takes place due to hardware or software issues of one or more nodes. Therefore, Crash-fault tolerance algorithms aim at high availability of the network by dividing the network into faulty and non-faulty nodes to keep the network functionality. On the other hand, byzantine failures take place due to a software bug, a malicious activity or a system compromise that causes one or more nodes to behave abnormally. Therefore, Byzantine algorithms aim

at availability too, but they can also detect the arbitrary nodes. They can divide the faulty machines into ill-functioning machines and malicious ones. BFT are developed with two major verification types: traditional block-based linear verification and vertices DAG-based verification. The common examples of each categories are explained in this section. Table 1 presents a comparison between common voting-based algorithms in terms of default access type, membership, architecture, active platforms, domain of usage, data form and the achieved TPS (transactions per second).

4.2.1.Crash Fault Tolerance Consensus Algorithms

Crash fault tolerance algorithms focus on the failure of one or more processes resulting from connectivity problems, system crashes or even process errors and corruptions. Paxos, RAFT, Viewstamped replication, Chubby and zookeeper are common examples of crash fault-tolerant consensus. In this section, we briefly discuss the most common CFT algorithms.

4.2.1.1. Paxos

Paxos[25] is significantly an old fault tolerance algorithm that is widely used over the last decades to ensure process continuity in case of failures among a group of distributed systems[26]. In multiple systems that propose values, paxos ensures that only one value is selected among the proposed values. Google's Spanner database[27] uses Paxos algorithm for its state machine replication to achieve replications among distributed DB Partitions. There are three main agents in paxos: proposers, acceptors, and learners. The proposer sends a proposed value to a set of acceptors with a unique identifier number. Acceptors may or may not accept the value based on the majority of agents who have accepted this value. Thus, the value is accepted if a large set of acceptors have accepted it. Zookeeper atomic Broadcast is a custom Paxos version that is implemented in Zookeeper platform.

4.2.1.2. Raft

RAFT[28] is an abbreviation of Reliable, Replicated, Redundant, and Fault-Tolerant which is a crash fault tolerance ordering service that was introduced as an update to paxos[28]. It differs from paxos in the idea of decomposing real systems problem into relatively independent sub-problems to be more practical, precise and understandable. Raft employs five server-nodes where two can crash at the same time. There are two separate parallel operations: Leader election and Log replication. Each server can work with a state of three: leader, follower and candidate. One leader is elected in each term, which handles all requests, while all other servers entitled the follower state. As seen in figure 6, the leader handles log sharing with the followers and periodically sends heartbeats to inform them of its existence. Each follower has a timeout in which it expects receiving heartbeat message from the leader. This timeout is reset once a heartbeat is received otherwise if the follower does not receive a heartbeat message within its timeout the follower changes its state to a candidate and starts leader election. Raft is used in Hyperledger fabric and Oracle blockchain platforms. Consul and etcd are known platforms that apply RAFT consensus.

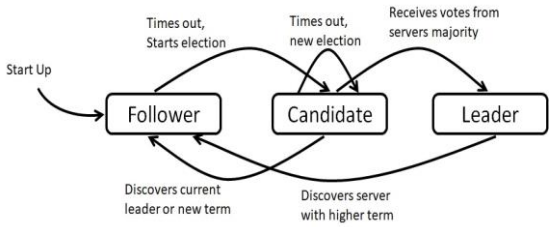


Figure 5: Raft three main servers' roles

4.2.2.Byzantine Fault Tolerance Consensus Algorithms

In distributed systems, the byzantine Generals problem takes place when the involved components must agree on a single strategy to avoid total failure due to false information or abnormal behavior from one or more nodes. The aim of the BFT-based consensus is to solve the known byzantine generals' problem in distributed systems[29] by avoiding system failure, corruption or malicious activity. BFT consensus allows the chained network to tolerate a certain number of bad simultaneous actors since they do not exceed the specified threshold (typically one-third of the total networked nodes in most BFT-based implementations). Thus, the higher number of participants, the more secure network. BFT also prevents forking since the committed blocks are final. BFT is referred as the atomic broadcast since the transaction is an atomic operation on a database that may either be completed or doesn't occur at all, but it can't kept in an intermediate state[30]. Below is a quick review about the most common BFT algorithms.

4.2.2.1.pBFT

Practical Byzantine Fault Tolerance algorithm was first introduced in 1999[31] to defend distributed systems in asynchronous network against malicious attacks and software errors. The algorithm presented the "state machine replication" mechanism that tolerates byzantine faults with up to [n-1/3] faulty simultaneous replicas out of total n replicas. pBFT divides the network nodes into two types: Primary and secondary nodes. Each node is assigned the "primary" state in a round robin fashion. A "view" represents the duration of time where a node holds the "primary" state. The primary node should perform the requested service such appending the new block during his "view" period. Nodes circulate the primary state according to an ordered node list dynamically generated based on a customized network configuration. PBFT is a semi-trusted consensus because only selected nodes can form the consortium. Hyberledger Fabric and Sawtooth are platforms that implement pBFT consensus to address the problem of whether to accept or ignore a piece of information from a leader.

4.2.2.2.dBFT

Delegated Byzantine Fault Tolerance is implemented in NEO blockchain system for large-scale participation in consensus through proxy voting Network [32]. dBFT clients are split into two different types: Bookkeepers (delegates) and ordinary (citizens). Bookkeeper are the elected participants based on BFT to apply the validation processes while ordinary nodes vote for the selection of bookkeepers. In order to become delegate, a node should hold a predefined amount of tokens based on a proof of stake algorithm. Ordinary nodes can vote regardless the amount of stake they possess. A "Speaker" node is randomly elected from bookkeepers to propose the new block. Delegates track citizen's transactions and store them

locally in a ledger after verification. To verify a new block, the speaker send his proposed block to all delegates to match with their own proposed blocks. Then, the new block is added if the minimum amount of citizens reached agreement about the block validity. If the amount of approvers do not reach the amount, a new speaker is elected and validation restarts. Voting in the NEO network handles up to 1000 TPS.

4.2.2.3. PoV

Proof of Vote[33] is developed for consortium blockchain to provide security, reliability, reduced verification delay time, less power consumption and to mitigate bifurcation. Different security identities are established by assigning each network member one of four roles: commissioner, Butler, butler candidate, and ordinary user. Commissioner is a machine that represents one of the members of the league committee, which must be accepted by the alliance law. To separate between voting rights and execution rights by design, butler is identified to produce blocks but they do not waste computing power in order to produce blocks. They are randomly selected to gather transaction from network packing them into blocks. To become butler, butler candidate undergoes voting by all commissioners to become butler in iterative elections. The block is marked valid and is added to the chain only after receiving at least 51% of the commissioners' votes.

4.2.2.4. PoT

Proof of trust was introduced to resist Sybil and collusion attacks. Nodes in POT are divided into four types; Leader nodes, gateway nodes, ledger management nodes, and validator nodes. POT undergoes four consecutive stages, where a leader is elected for ledger management in the first stage. Then, the elected leader choses a transaction validation group based on custom voting mechanism. Thirdly, that validation group vote for which transaction shall be added to the next block. Finally, the approved transaction is added to the chain. It takes about just 4 seconds to append a new valid block to the chain making POT better in performance, scalability and agreement time[34].

4.2.2.5. Tendermint

Tendermint is a protocol for ordering events in a distributed network that is inspired by PBFT algorithm[35]. Hence, it allows less than one-third distributed faulty processes of the total voting power. Tendermint aims at achieving thousands of transactions per second on dozens of nodes with very high performance and one-second latency. It targets distributed IoT devices across the entire world in terms of performance and protection against cyber-attacks. Tendermint platform is implemented in Go and first introduced in 2015 with a custom interface for building IIOT arbitrary applications. The algorithm and a set of deployment tools are open source codes located at GitHub[36]. It utilizes the gossip protocol for exchanging messages between processes where both received and sent messages are stored in a local message log for every process. It is deployed in the form of a state machine replication[37] that adds combined transactions in blocks.

4.2.2.6. Corda

Corda is a permissioned distributed ledger platform made of mutually distrusting nodes that allows a single global database to record the state of agreements between people and institutions[38]. It provides a P2P communication on a need-to-know basis. The main intention of Corda is to deploy financial smart

contracts by allowing a greater level of code sharing facility for financial industries so that it reduces the cost of financial services. The two major features of Corda are automated smart contracts and time stamping of documents. Corda does not need every node in the network to hold a copy of the entire ledger. There is no single network-wide ledger. Nodes in Corda only need to agree on shared facts relevant to them. In the absence of a single global view of the ledger, nodes use a process called "walking the chain" to verify the provenance of assets being consumed in a Corda transaction. All changes in Corda are made through transactions, which change the state object of assets based on the contracted logic.

4.2.2.7. Hashgraph

A hashgraph is a data structure developed by Leemon Baird in 2016 which records gossips between peers and the order of these gossip-events[16]. Gossip is the information shared by a member with other members that he selects randomly. Network members can create signed transactions at any time and allow others to apply byzantine agreement to validate the order of these transactions. The hashgraph is distributed through the gossip protocol maintaining the history of previous gossips. Nodes build a hashgraph reflecting all of the gossip events. An event is a small data structure in memory that is digitally signed by its creator. The graph keeps growing making older parts immutable. Each member repeatedly calls other members at a predefined time intervals to synchronize its gossip. These time intervals are called rounds and the first event that a member creates in each round is called a witness. Witnesses can be either famous or not famous. The fame of each witness is calculated by considering how many witnesses linked to it in the next round. There must be enough different paths to an event through the majority of the population. Hedera-Hashgraph, IOTA and Railblocks are common DAG-based implementations[39].

4.2.2.8.Tangle

An open source blockless-DLT based on DAG[40]. It offers huge scalable networks with the lowest cost. It does not package transactions in blocks, nor do stores block hash to track order of transactions. Transactions are stored on multiple devices distributed across various locations. Transaction are approved when the issuer verifies two other unapproved transactions called tips in the ledger. Thus, each added transaction requires selecting two random unapproved transactions after getting a confirmation confidence. Confirmation confidence allow the system to validate new transaction and measure how much it is acceptable by the ledger based on tips counting[41]. Tangle uses SHA3 to resist quantum attacks. IOTA is an example which released in 2016 to support cryptocurrencies for industrial IoT[42]. Uber is a use case for tangle that generates thousands of micro-transactions and data records.

4.2.2.9.SCP

Stellar Consensus Protocol is a quorum-based BFT open source decentralized permission-less open-membership consensus algorithm that was produced in 2015[43]. Stellar applies the FBA (Federated Byzantine Agreement) rather than traditional BFT. It adopts the replicating state machine approach to validate a global ledger but it differs from other BFA that it gains its trust from an internet-level consensus rather than a predefined validator list. Each validators group decides which other validators to trust. Each group of validators are called a Quorum slice where quorum slices of each validator may overlap to form a quorum or network wide consensus. It was introduced to resist distributed Sybil attacks[44]. Stellar uses the "message passing" protocol that requires significantly less power for confirming transactions. Therefore, it allows thousands of TPS with very low cost in a complete decentralized network. Based on

the FLP impossibility proof[45], Stellar prefers safety to liveness. Hence, if there were a disconnection in the internet SCP would hold the progress of the network until consensus occurs through the quorum. On the contrary, liveness-preferring blockchains would proceed in each isolated network section leading to forking.

4.2.2.10. RPCA

Ripple Protocol Consensus is an open source low-latency Byzantine fault-tolerant agreement Algorithm that was introduced for distributed payment networks. The Ripple platform allows international digital payment and asset secure transfer between different platforms that trade with different cryptocurrencies. It uses a native cryptocurrency token called XRP[46] that acts as an intermediate form of exchange between different monetary systems that trade with different cryptocurrencies. RIPPLE aims at solving the three main problems of distributed payment systems, which are correctness, agreement and utility[47]. RIPPLE tolerates up to 1/5 faulty nodes of the distributed network[48]. There is no global knowledge of all participants in RIPPLE network, but each node declares a list of other nodes that it trusts and consider for voting. Similar to Stellar, RIPPLE follows the FBA technique. RIPPLE has some security issues declared in [49] and [48] such as double spending, Sybil attacks and poor scalability.

Table 1: Comparison between Voting-Based Consensus Protocols

	Class	Access Type	Membership	TPS	Node Classes	Common Platforms	Domain	Data form
Paxos	Crash	Public	Open	Instantly	Three Types Proposers, Acceptors, Learners	Zookeeper	Monetary, Iot, Smart Contracts, DApps	Block & Blockless
Raft		Public	Open-Closed	Instantly	Three Types Leader, Follower, Candidate	Hyperledger Fabric, etcd, Consul	IoT, Distributed Apps	Block & Blockless
PBFT	BFA	Private Consortium	Closed	Low	Two types Primary, secondary	Sawtooth, Hyberledger	IoT, distributed Apps	Blocks
DBFT		Private Consortium	Closed	15:20S for block	Two types Bookkeeper, Ordinary	NEO (Ethereum of China)	Cryptocurrency	Blocks
PoV		Consortium	Closed	60 TPS[50]	Four types commissioner, Butler, Butler candidate, Ordinary	Chainspace	Smart contracts, DApps	Blocks
PoT[34]		Private Consortium	Closed	Low	Four Types Leaders, Followers, Gateway, Validating	CrowdBC PrivCrowd	Crowdsourcing[51]	Blocks
Tendermint		Private Consortium	Closed	Very low	Two types Proposer, Ordinary	Tendermint	Distributed applications	Blocks
Corda		Private Consortium	Closed	170 TPS	One type All nodes can Validate	Corda R3	Industrial Distributed Applications	Blockless
HashGraph		Public	Open	250000 TPS	One type (Same voting level)	Swirlds, Hedera	Distributed applications	Blockless

Tangle		Public	Open-Closed	800 TPS	One type (Same voting level)	IOTA	IoT, Digital Payments	Blockless
SCP	FBA	Public	Open	3:5 seconds	One Type	Stellar	Global Money Transfers	Blocks
RPCA		Public	Open	Few seconds	Two types Server, Client	RIPPLE	Cryptocurrency Exchange	Blocks

5. Selection Criteria for IIoT Consensus Algorithms.

There are few DLT-based solutions introduced for industrial IoT. In order to produce such a protocol that provides fault tolerance, real-time transactions, and native security with relatively low computing and energy-consumption, researchers and developers endeavor to produce new algorithms customized for IIoT or significantly modify the existing legacy consensus algorithms. Ethereum, Corda and HyberLedger Fabric are common platforms that introduce modified custom variants of older blockchain consensus to meet the IIoT. On the other hand, Tangle, Tendermint and IOTA are examples of platforms that introduced primarily for Industrial IoT. While stellar is made for money exchange, but its architecture can be customized to accommodate IIoT applications. In this context, we define the major criteria that govern the development of a DLT solution suitable for IIoT. Table 2 presents an overall comparison between the main categories of consensus. It's stated in [52], that voting-based consensus is preferred for IoT rather than proof-based consensus and also recommended private DLT than public. However, [23] and [53] introduced proof-based consensus protocols and claimed they meet IoT requirements in terms of wait time, fairness and resource consumption. According to[17] byzantine-based protocols are generally inadequate for large-scale network and require significant adaptation to cope with IoT. Thus, we discuss here the main criteria to consider by industrial community to build their adequate business related IIoT-DLT as illustrated in figure 7.

Table 2: Comparison between the Main Categories of Consensus

	Proof-Based Consensus	**Voting Based Consensus**		
		Byzantine Generals Problem		**Crash Fault Tolerance**
		BFA	FBA	
Blockchain Type	Public	Private	Public	Public
Membership	Open	Closed	Open	Open
FLP Preference	Liveness	Safety	Safety	Liveness
Hash-Based	✓	×	×	×
Validation Process	Mining	Leader Election	Leader Election	Server Election
Data Verification	Block-Based/Linear	Linear / Vertices (Ordered or DAG)	Linear (Ordered Records)	Linear (Ordered Records)
Transaction Speed	Slow	Fast	Fast	Very fast
Resource Greediness	High	Low	Low	Low
Asymptotic Security	Not considered	Considered	Considered	Low Consideration

5.1. Constrained Sensing Devices

Constrained devices limitations are resolved by one of two options. The first is to develop two different software codes, one for fully functioning powerful nodes and the other for lightweight nodes. Full nodes host a complete ledger and apply all verification functions while lightweight nodes merely send their

operational data. This allows constrained devices to run partial codes or act as verified clients without participating in the agreement processes. The second option is to develop a complete lightweight blockchain technique suitable to run on low-resources nodes. The choice between two options is based on the capabilities of most nodes. The first option is recommended for DLT that comprises many powerful nodes, while the second option is preferred for DLT with majority of constrained devices. Authors of [54] presented an optimized PoV consensus for constrained devices. IOTA foundation[55] also presented "Hornet" virtual node for low resources nodes. Some implementation like [56] uses a reverse-proxy gateway for data communication between sensor network and Public IoT-chained network.

5.2. Population Size, Fault Tolerance and Scalability

IoT represents millions of distributed connected devices across the universe that feeds each other's with valuable information. When validating this information through blockchain, some devices (if not all) should carry out the effort of data validation and provide adequate storage to store the ever-expanding ledger. Generally, "The higher number of legitimate devices, the more security and availability for the chained network". The smaller number of DLT nodes the more attacking risk by lower computing machines" [57]. However, more activities between devices consumes more computing power and storage. Thus, byzantine problem based protocols are inadequate for large-scale network[17] since they generate very large number of voting messages that is hard for a network to handle. Similarly, fault tolerance and business continuity increases proportional to population size. Scalability is also an issue for always expanding industries or those willing to apply internet-based consensus because the more events between devices, the more computing power and storage required.

5.3. DLT Type and Membership

Traditional non-industrial networks allow communications between users and operational applications. Thus, they prefer public blockchain for their applications to ensure operability and wider communication space. However, smart factories and manufacturing plants do not have interest towards public blockchains. Public blockchained-network strengthens transparency, trust and data validation but weakens privacy and increase the risk of unauthorized access. Manipulating data records or compromising critical manufacturing equipment may cause tremendous risks such as systems disruption, production shutdown, environmental disaster, and human injuries or even death. Therefore, permissioned private or consortium blockchains are preferable to open public blockchains in this case.

5.4. Confirmation Time and network Delay

Critical industries apply advanced autonomous systems that require real-time transactions to take accurate and on-time decisions based on the communicated messages. Legacy proof-based consensus does not satisfy the required continuity of critical data streams. On the contrary, voting based consensus proved their effectiveness in terms of faster transaction confirmation, low latency and large number of TPS since there is no mining processes.

5.5. FLP Preferences

The safest environment does not accept any changes in its state while the most live environment accepts all changes. In other words, accepting all transactions extremely satisfies liveness and accepting no transactions extremely satisfies safety. Thus, violating safety takes place when two or more competing or

contradicting changes are accepted while violating liveness takes place when the network stop responding to changes. It is a trade-off decision to balance between liveness and safety[45]. Thus, Consistency-oriented algorithms focus on keeping the same values at all replicas at the same time while resilience is to guarantee the strength of the network and its functionality in case of failures to some of its participants. The more failing nodes in a running network, the higher resilience level. While the more nodes that terminate with the same results and same state, the more consistency level. Paxos-based algorithms provide continuity so that the network keeps going in case of failures for some nodes. View-based algorithms care much for security against functionality so that a network may stop until the corrupted node or network partition comes back to normal.

5.6. Communication Protocols

Communication in distributed systems may be synchrony or asynchrony. Synchrony protocols entail predefined fixed values for delays and time intervals, while asynchrony protocols have no timing limits. Although synchronous model reduces uncertainty, but it is not practically applied in any of the current DLT platforms due to the nature of the internet connection that may takes longer durations to define failures. In DLT, control messages are interchanged between participating nodes regularly to vote for a leader, check a leader availability or to append new data to the ledger. The message passing protocol allows servers to exchange periodic messages to indicate its current state based on all-to-all pattern, which may affect the performance in large-scale networks. Alternatively, GOSSIP[35] protocol used in DAG-based consensus is repeatedly applies calls between involved members only at a predefined time intervals to synchronize their gossips. Atomic broadcast is also a communication technique similar to consensus designed mainly to enable highly available database services that allow distributed systems agree on the same ordered sequence of messages. In atomic broadcast, operation on a database may either be completed or doesn't occur at all, but it can't kept in an intermediate state[30]. Ordering is performed by tagging messages with timestamps that help in ordering and therefore, must contain information on every server. Most practical atomic broadcast algorithms were designed mainly to enable highly available distributed services.

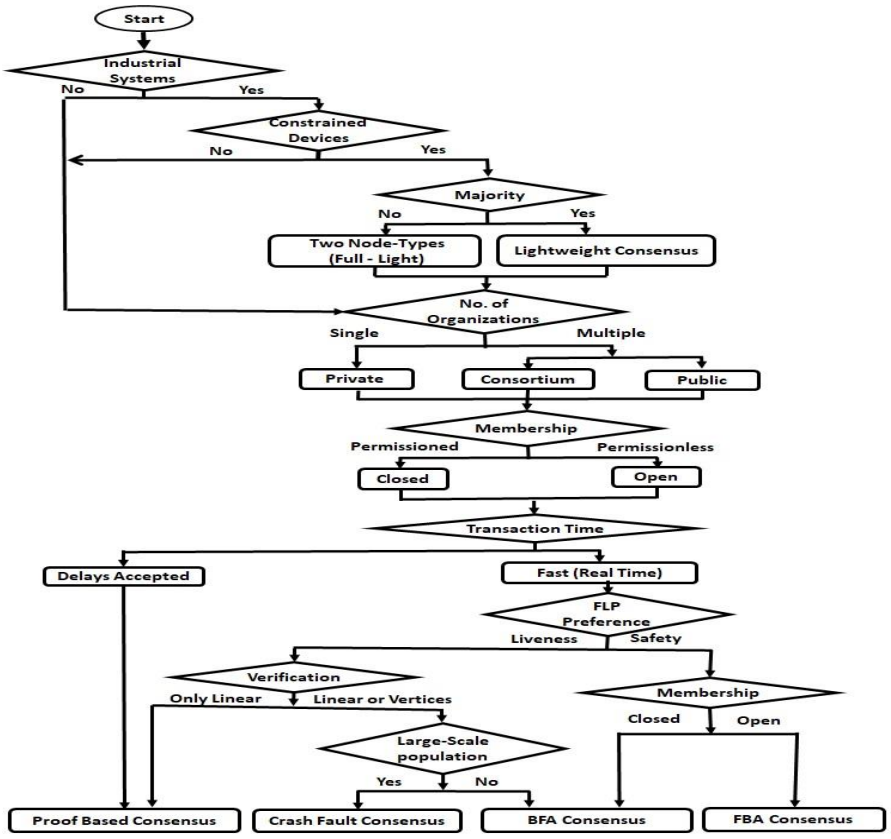


Figure 6: Selection Chart for DLT Consensus

6. Conclusion and Future work

The use of DLT for securing industrial IoT is expected to flourish very fast within the next few years. Voting-based protocols proved higher compatibility with industrial IoT rather than proof-based protocols in terms of protection and performance. Byzantine fault tolerance protocols based on DAG verification can solve many challenges of large networks in terms of scalability, security and resource greediness than linear-based verification. This work introduced DLT as a key component of industrial IoT focusing on its main advantages when integrated with IoT for industrial environments. It also surveyed common DLT consensuses providing comparative analysis between the most common algorithms. This survey helped to draw a selection chart that highlights the main criteria to consider for building a DLT solution for different industries. The main objective of the study is to help the industrial community in adopting their business-relevant consensus protocol and build appropriate DLT. This study can pave the way for a future open industrial DLT model that works as a backbone for all types of IIoT to ensure distributed secure manufacturing.

References

[1] I. T. Christou, N. Kefalakis, A. Zalonis, J. Soldatos, and R. Bröchler, "End-to-End Industrial IoT Platform for Actionable Predictive Maintenance," IFAC-PapersOnLine, vol. 53, no. 3, pp. 173-178, 2020.

[2] IDC, "IoT Growth Demands Rethink of Long-Term Storage Strategies," IDC Release Note. 27 July 2020, p. https://www.idc.com/getdoc.jsp?containerId=prAP46737220, 2020.
[3] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," Future generation computer systems, vol. 82, pp. 395-411, 2018.
[4] M. Yu, J. Zhuge, M. Cao, Z. Shi, and L. Jiang, "A survey of security vulnerability analysis, discovery, detection, and mitigation on IoT devices," Future Internet, vol. 12, no. 2, p. 27, 2020.
[5] D. Kiel, C. Arnold, and K.-I. Voigt, "The influence of the Industrial Internet of Things on business models of established manufacturing companies–A business level perspective," Technovation, vol. 68, pp. 4-19, 2017.
[6] S. Li, S. Zhao, P. Yang, P. Andriotis, L. Xu, and Q. Sun, "Distributed consensus algorithm for events detection in cyber-physical systems," IEEE Internet of Things Journal, vol. 6, no. 2, pp. 2299-2308, 2019.
[7] S. Zhao, S. Li, and Y. Yao, "Blockchain enabled industrial Internet of Things technology," IEEE Transactions on Computational Social Systems, vol. 6, no. 6, pp. 1442-1453, 2019.
[8] L. Da Xu, W. He, and S. Li, "Internet of things in industries: A survey," IEEE Transactions on industrial informatics, vol. 10, no. 4, pp. 2233-2243, 2014.
[9] Z. Shi, Y. Xie, W. Xue, Y. Chen, L. Fu, and X. Xu, "Smart factory in Industry 4.0," Systems Research and Behavioral Science, vol. 37, no. 4, pp. 607-617, 2020.
[10] E. Hozdić, "Smart factory for industry 4.0: A review," International Journal of Modern Manufacturing Technologies, vol. 7, no. 1, pp. 28-35, 2015.
[11] V. Dieterich, M. Ivanovic, T. Meier, S. Zäpfel, M. Utz, and P. Sandner, "Application of blockchain technology in the manufacturing industry," Frankfurt School Blockchain Center, Germany, pp. 1-23, 2017.
[12] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system.(2008)," ed, 2008.
[13] I. Makarov and A. Schoar, "Trading and arbitrage in cryptocurrency markets," Journal of Financial Economics, vol. 135, no. 2, pp. 293-319, 2020.
[14] S. Nakamoto, "NewBull: A Peer-to-Peer Electronic Cash System."
[15] F. B. Schneider, "Implementing fault-tolerant services using the state machine approach: A tutorial," ACM Computing Surveys (CSUR), vol. 22, no. 4, pp. 299-319, 1990.
[16] L. Baird, "The swirlds hashgraph consensus algorithm: Fair, fast, byzantine fault tolerance," Swirlds Tech Reports SWIRLDS-TR-2016-01, Tech. Rep, 2016.
[17] R. Han, V. Gramoli, and X. Xu, "Evaluating blockchains for IoT," in 2018 9Th IFIP international conference on new technologies, mobility and security (NTMS), 2018, pp. 1-5: IEEE.
[18] G.-T. Nguyen and K. Kim, "A survey about consensus algorithms used in blockchain," Journal of Information processing systems, vol. 14, no. 1, pp. 101-128, 2018.
[19] Y. Wen, F. Lu, Y. Liu, and X. Huang, "Attacks and countermeasures on blockchains: A survey from layering perspective," Computer Networks, vol. 191, p. 107978, 2021.
[20] S. King and S. Nadal, "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake," self-published paper, August, vol. 19, p. 1, 2012.
[21] D. Larimer, "Delegated proof-of-stake (dpos)," Bitshare whitepaper, vol. 81, p. 85, 2014.
[22] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, "Proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract] y," ACM SIGMETRICS Performance Evaluation Review, vol. 42, no. 3, pp. 34-37, 2014.
[23] K. Karantias, A. Kiayias, and D. Zindros, "Proof-of-burn," in International Conference on Financial Cryptography and Data Security, 2020, pp. 523-540: Springer.

[24]B. Ampel, M. Patton, and H. Chen, "Performance modeling of hyperledger sawtooth blockchain," in 2019 IEEE International Conference on Intelligence and Security Informatics (ISI), 2019, pp. 59-61: IEEE.
[25]L. Lamport, "The part-time parliament," in Concurrency: the Works of Leslie Lamport, 2019, pp. 277-317.
[26]L. Lamport, "Paxos made simple," ACM Sigact News, vol. 32, no. 4, pp. 18-25, 2001.
[27]J. C. Corbett et al., "Spanner: Google's globally distributed database," ACM Transactions on Computer Systems (TOCS), vol. 31, no. 3, pp. 1-22, 2013.
[28]D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in 2014 {USENIX} Annual Technical Conference ({USENIX}{ATC} 14), 2014, pp. 305-319.
[29]L. Lamport, R. Shostak, and M. Pease, "The Byzantine Generals Problem ACM Transactions on Progamming Languages and Syetems, vol. 4 no. 3 pp. 382-401," ed: July, 1982.
[30]J. Gray, "The transaction concept: Virtues and limitations," in VLDB, 1981, vol. 81, pp. 144-154.
[31]M. Castro and B. Liskov, "Practical byzantine fault tolerance," in OSDI, 1999, vol. 99, no. 1999, pp. 173-186.
[32]A. Burnie, "Exploring the interconnectedness of cryptocurrencies using correlation networks," arXiv preprint arXiv:1806.06632, 2018.
[33]K. Li, H. Li, H. Hou, K. Li, and Y. Chen, "Proof of vote: A high-performance consensus protocol based on vote mechanism & consortium blockchain," in 2017 IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2017, pp. 466-473: IEEE.
[34]J. Zou, B. Ye, L. Qu, Y. Wang, M. A. Orgun, and L. Li, "A proof-of-trust consensus protocol for enhancing accountability in crowdsourcing services," IEEE Transactions on Services Computing, vol. 12, no. 3, pp. 429-445, 2018.
[35]E. Buchman, J. Kwon, and Z. Milosevic, "The latest gossip on BFT consensus," arXiv preprint arXiv:1807.04938, 2018.
[36]https://github.com/tendermint/tendermint. (2015). https://github.com/tendermint/tendermint. Available: https://github.com/tendermint/tendermint
[37]E. Buchman, "Tendermint: Byzantine fault tolerance in the age of blockchains," 2016.
[38]R. G. Brown, J. Carlyle, I. Grigg, and M. Hearn, "Corda: an introduction," R3 CEV, August, vol. 1, p. 15, 2016.
[39]L. Baird, M. Harmon, and P. Madsen, "Hedera: A public hashgraph network & Governing Council," White Paper, vol. 1, 2019.
[40]P. Schueffel, "Alternative distributed ledger technologies Blockchain vs. Tangle vs. Hashgraph-A high-level overview and comparison," Tangle vs. Hashgraph-A High-Level Overview and Comparison (December 15, 2017), 2017.
[41]S. Popov, "The tangle," White paper, vol. 1, no. 3, 2018.
[42]B. Cao et al., "When Internet of Things meets blockchain: Challenges in distributed consensus," IEEE Network, vol. 33, no. 6, pp. 133-139, 2019.
[43]D. Mazieres, "The stellar consensus protocol: A federated model for internet-level consensus," Stellar Development Foundation, vol. 32, 2015.
[44]N. Barry, G. Losa, D. Mazieres, J. McCaleb, and S. Polu, "The Stellar Consensus Protocol (SCP)," Internet Engineering Task Force, Internet-Draft draft-mazieres-dinrg-scp-05, 2018.
[45]E. Borowsky and E. Gafni, "Generalized FLP impossibility result for t-resilient asynchronous computations," in Proceedings of the twenty-fifth annual ACM symposium on Theory of computing, 1993, pp. 91-100.

[46]B. Chase and E. MacBrough, "Analysis of the XRP ledger consensus protocol," arXiv preprint arXiv:1802.07242, 2018.
[47]D. Schwartz, N. Youngs, and A. Britto, "The ripple protocol consensus algorithm," Ripple Labs Inc White Paper, vol. 5, no. 8, p. 151, 2014.
[48]I. Amores-Sesar, C. Cachin, and J. Mićić, "Security Analysis of Ripple Consensus," arXiv preprint arXiv:2011.14816, 2020.
[49]P. Todd, "Ripple protocol consensus algorithm review," May 11th, 2015.
[50]Y. Abuidris, R. Kumar, T. Yang, and J. Onginjo, "Secure large-scale E-voting system based on blockchain contract using a hybrid consensus model combined with sharding," Etri Journal, vol. 43, no. 2, pp. 357-370, 2021.
[51]M. Li et al., "Crowdbc: A blockchain-based decentralized framework for crowdsourcing," IEEE Transactions on Parallel and Distributed Systems, vol. 30, no. 6, pp. 1251-1266, 2018.
[52]M. Salimitari and M. Chatterjee, "A survey on consensus protocols in blockchain for iot networks," arXiv preprint arXiv:1809.05613, 2018.
[53]M. Milutinovic, W. He, H. Wu, and M. Kanwal, "Proof of luck: An efficient blockchain consensus protocol," in proceedings of the 1st Workshop on System Software for Trusted Execution, 2016, pp. 1-6.
[54]A. D. Dwivedi, L. Malina, P. Dzurenda, and G. Srivastava, "Optimized blockchain model for internet of things based healthcare applications," in 2019 42nd international conference on telecommunications and signal processing (TSP), 2019, pp. 135-139: IEEE.
[55]M. Bhandary, M. Parmar, and D. Ambawade, "A blockchain solution based on directed acyclic graph for IoT data security using IoTA tangle," in 2020 5th International Conference on Communication and Electronics Systems (ICCES), 2020, pp. 827-832: IEEE.
[56]K. O.-B. Obour Agyekum et al., "A secured proxy-based data sharing module in IoT environments using blockchain," Sensors, vol. 19, no. 5, p. 1235, 2019.
[57]J. J. Xu, "Are blockchains immune to all malicious attacks?," Financial Innovation, vol. 2, no. 1, pp. 1-9, 2016.