*Article*

# A Review of Blockchain Architecture and Consensus Protocols: Use Cases, Challenges, and Solutions

**Leila Ismail [1],* and Huned Materwala [1]**

[1] College of IT, UAE University, Al Ain, United Arab Emirates
* Correspondence: leila@uaeu.ac.ae

**Abstract:** Over the last decade the blockchain technology has emerged to provide solutions to the complexity and privacy challenges of using distributed databases. Its adoption is motivated by cost reduction for customers. Over this time, the concept of blockchain has shifted greatly due to its potential in business growth for enterprises and the rapidly evolving applications in a collaborative smart cities ecosystem, healthcare, and governance. Many platforms, with different architectures and consensus protocols, have been introduced. Consequently, it becomes challenging for an application developer to chose the right platform. Furthermore, blockchain has misaligned with the goals for efficient green collaborative digital ecosystem. Therefore, it becomes critical to address this gap and to build new frameworks to align blockchain with those goals. In this paper, we discuss the evolution of blockchain architecture and consensus protocols, bringing a retrospective analysis and discussing the rationale of the evolution of the various architectures and protocols, as well as capturing the assumptions conducting to their development and contributions to building collaborative applications. We introduce a classification of those architectures, helping developers to chose a suitable platform for applications and providing insights for future research directions in the field in building new frameworks.

**Keywords:** Blockchain, Consensus, Hash functions, Privacy, Replication, Scalability.

## 1. Introduction

The blockchain is a disruptive technology, emerged for decentralized applications as the outcome of painful complexity, security, and intermediary extending across over half a century. Blockchain, a peer-to-peer system, enable users to maintain a ledger of transactions that is replicated and synchronized over multiple users' servers [1]. The transactions are processed and verified by consensus of a majority of the network participants eliminating the need of an intermediary. The transactions are packed in blocks and the blocks are chained together using a cryptographic hash to provide immutability. Since its introduction in 2008 [2], the blockchain platforms and consensus protocols have proliferated, due to the evolution of collaborative applications in smart cities, such as healthcare and governance, as well as the need of green and cost-efficient computing. Therefore, it becomes difficult for an application developer to chose the right platform. In addition, current blockchain architectures and consensus protocols have misaligned with the goals for a green collaborative decentralized and agile ecosystem. As such, it is becoming more and more vital to address this issue, and build new frameworks to align blockchain with those goals. Our main goal in this paper is to help developers to chose the right platform architecture and consensus protocol for applications. Therefore, we classify the platforms architectures into categories based on the applications nature in terms of number of ledgers and interoperability needs. This is to help applications' developers to chose the right platform. We discuss the advantages of each category and inherent problems therein, and begin offering solutions towards a scalable, cost efficient, and green blockchain framework.

Originally developed for transfer of digital currency without relying on intermediaries [2], blockchain has evolved to serve decentralized applications. With the rise of collaborative ecosystems for better customer services, and the enormous amount of energy consumed by the underlying blockchain architecture and consensus protocol, it becomes more and more difficult to foresee the uses and applications of blockchain, therefore compounding the difficulty in achieving the goals. In 2017, the bitcoin mining used around 30.14 TerraWatt hours (TWh) of energy, which is equivalent to energy usage of the entire Ireland in a year [3]. According to a research, the annual carbon dioxide emissions by the bitcoin network are as high as 22.9 million metric tons, almost equivalent to the amount produced by the countries like Sri Lanka and Jordan [4].

Further challenges have been placed on distributed applications by the expanding industrial market growth to serve a wide number of customers. Growing business requires trust and transparency between the customers and business providers. Customers require to eliminate intermediaries to reduce transactions cost. The issues of data communication overhead with increasing number of network participants further hinders the real-time services of the developed applications. Addressing these problems of energy consumption and scalability often trades off with security, and privacy. As such, the goals of this paper are four-fold: 1) we aim to provide a temporal evolution of blockchain applications development platforms architectures and consensus protocols with a retrospective analysis to their introduction. We classify the platforms and the consensus protocols under unifying architectures, and discuss various existing and upcoming blockchain applications, 2) help developers to choose the right platform architecture for applications, 3) we evaluate the current research on the topic, blockchain architecture and consensus protocols under comprehensive taxonomies, and address the challenges and issues in that respect, and 4) we utilize the taxonomies to guide future directions in the field.

The main contributions of the paper are as follows.

1. We present an overview of the blockchain layers and its transaction execution and data flow which are common to all blockchain architectures.
2. We classify blockchain platforms architectures into three different types based on the nature of applications using them in terms of number of ledgers and interoperability. A taxonomy of different blockchain architecture mapped to corresponding development platforms is also presented.
3. We present the scalable characteristics of each architecture and the security techniques employed.
4. We introduce a taxonomy, classification and comparison between the different consensus protocols used in the blockchain literature.
5. We describe different existing and upcoming blockchain applications in the domain areas such as medical, finance, education, manufacturing industry and retail marketplace, media, real estate, transportation, government, authorship and ownership, and digital content management.
6. Based on this study, we provide critical analysis of the different issues prevailing in blockchain technology and the possible solutions that have been proposed for these issues, along with future research directions.
7. We propose directions toward the development of scalable, and energy-efficient to address the void between the existing blockchain architecture and consensus protocols and the services of the evolving applications.

To the best of our knowledge, we are the first to present a detailed systematic survey of blockchain along with a taxonomy of platforms architectures and 21 consensus protocols.

The rest of the paper proceeds as follows. Section 2 provides an overview of the related surveys. synthesizes a taxonomy of the works on VM placement. The overview of blockchain and transaction execution flow along with an organizational framework is presented in Section 3. Section 4 and 5 synthesizes the taxonomies of blockchain architectures and consensus protocols respectively. The existing blockchain applications and the potential of blockchain in various applications domains are described in Section 6. Section 7 highlights the issues prevailing in blockchain along with possible solutions for future research directions. The paper is concluded with possible future directions in Section 8.

## 2. Related Surveys

While there has been a lot of attention towards the blockchain technology, there are relatively few surveys conducted in this area. These survyes can be classified into one of the 3 categories: 1) applications [5–13], 2) privacy issues and security threats [14–16], and 3) consensus protocols [1,17–19].

Regarding applications, Ahamad et al. [5] presented the benefits and limitations of the different digital currencies for financial domain. Tschorsch et al. [6] discussed the process of transactions validation and block mining in a bitcoin network and the issues of scalability, security and privacy, anonymity, double spending, and pooled mining. Shen et al. [7] performed a literature review of different blockchain use cases for smart cities and organize those cases into nine different categories. The nine application categories are governance and citizen engagement, education, healthcare, economy, transportation, energy, water and waste management, civil construction, and natural environment. Jaroodi et al. [8] presented the benefits and challenges of using blockchain for financial, healthcare, logistics, manufacturing, energy, food and agriculture, robotics, construction, telecommunications, and entertainment applications. A similar study is carried out by Jaoude et al. [9]. Hölbl et al. [10] conducted a systematic review on the use of blockchain in healthcare. Similarly, Conoscenti et al. [11] discussed the issues of scalability, integrity and privacy in blockchain for Internet of Things (IoT) applications. Karafiloski et al. [12] presented the different blockchain solutions for storing and processing Big Data. Yli-Huumo et al. [13] presented a systematic review on the blockchain technology and its applications. The authors concluded that there is a research gap on blockchain architecture and scalability. In this paper, we present a temporal evolution of the different blockchain platforms architectures and capture the assumptions conducted to the development of a given platform architecture, revealing a retrospective analysis of their features over time. In addition, we present a taxonomy of blockchain architectures and give insights toward developing a new blockchain framework for building scalable and cost-efficient blockchain framework. We then map the development platforms to one of the classified architectures.

Concerning security threats and privacy issues, Li et al. [14] investigated different threats such as double spending, 51% attack, private key security, vulnerable and malicious activities, and data privacy in the blockchain network. Feng et al. [15] reviewed the different privacy threats and provided a taxonomy of existing cryptographic defense mechanisms to tackle those threats. Park et al. [16] presented the security challenges and associated solutions in the bitcoin network. The authors also proposed solution to secure data in cloud computing environment using Blockchain.

For consensus protocols, Mukhopadhyay et al. [17] surveys the mining techniques in consensus protocols used by the different cryptocurrencies. Zheng et al. conducted a survey of 6 consensus protocols [1] presenting their limitations in terms of scalability, privacy, and selfish mining for bitcoins. Wahab et al. conducted a survey of 7 consensus protocols [19] presenting their advantages and drawbacks. Nguyen et al. highlighted some consensus protocols [18] and classify them into proof of work, proof of stake and voting based. The classification is based on computing power usage, cryptocurrencies and number of votes. The classification is fine-grained which does not allow to add future non-cryptocurrencies-based consensus protocols. In this paper we provide the evolution over time of the consensus protocols that underpin the blockchain technology, seizing the rationals which led to the development of a given protocol. We then introduce a large-grain taxonomy for 21 consensus protocols including a capability-based category in addition to the computing power based and number of votes. Capabilities can be research contributions, storage, trust level, and cryptocurrencies.

## 3. Blockchain: An architecture perspective

In this section we describe a layered overview of blockchain and explain how a transaction data is processed in a blockchain network. We then present an organizational framework to help readers to effectively design blockchain architecture and to develop applications.

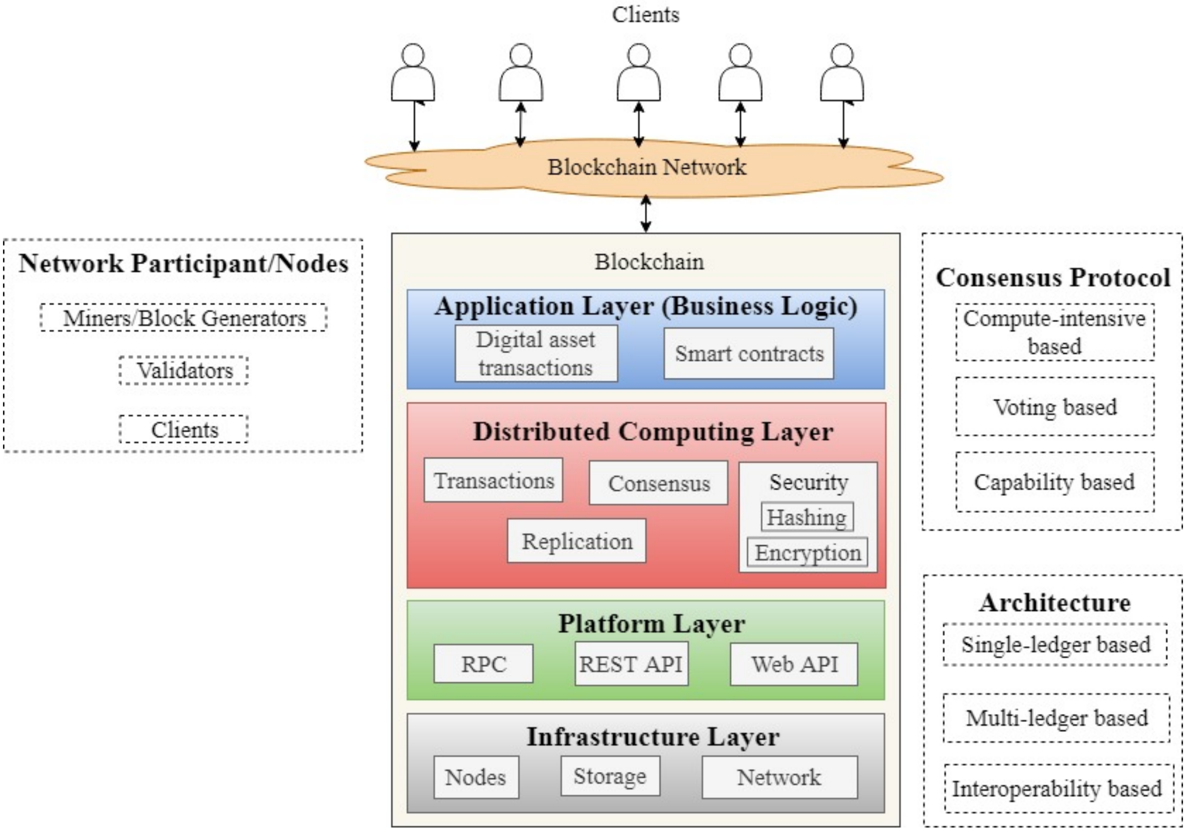

**Figure 1.** Overview of Blockchain.

### *3.1. Blockchain Overview*

Figure 1 shows an overview of blockchain technology. We divide the blockchain architecture in four layers: infrastructure, platform, distributed computing, and application. The infrastructure layer consists of all the hardware components required to run the blockchain, such as nodes, storage, and network facilities. The nodes are the network participants. A typical blockchain network has three different type of nodes: simple node (also referred to as a light node), full node and mining node. A simple node in the network can just send and receive transactions and does not store a copy of the ledger, neither validate a transaction, whereas a full node do. A Mining node (also referred to as a block generator) is a full node with the capability of mining, i.e., the process of generating a new block. The storage component stores the ledger of the transactions records. The platform layer facilitates remote procedure calls (RPC) [20], web application programming interface (API) [21], and REST API's [22] for the communication between the network participants.

The distributed computing layer ensures local access to the data, fault tolerance, immutability, privacy, authenticity, and security of the transaction data. Immutability is the property that does not allow modification of transaction records once stored in the ledger. The blockchain network uses a consensus protocol to reach an agreement regarding the order of the transactions in the network, the update of the ledger, and the selection of a miner for a block generation. In addition, this layer is responsible for user authentication using an encryption technique [23] and data privacy using a hashing technique [24]. The application layer is the business logic for applications such as digital asset transactions and smart contracts. An application developed on top of a blockchain network can be accessed by the clients using the platform layer. The blockchain architecture comprising of different layers has the following characteristics.

- Decentralization: The transactions in Blcokchain are processed and validated by the majority of the network participants based on consensus and are stored in a ledger replicated over multiple

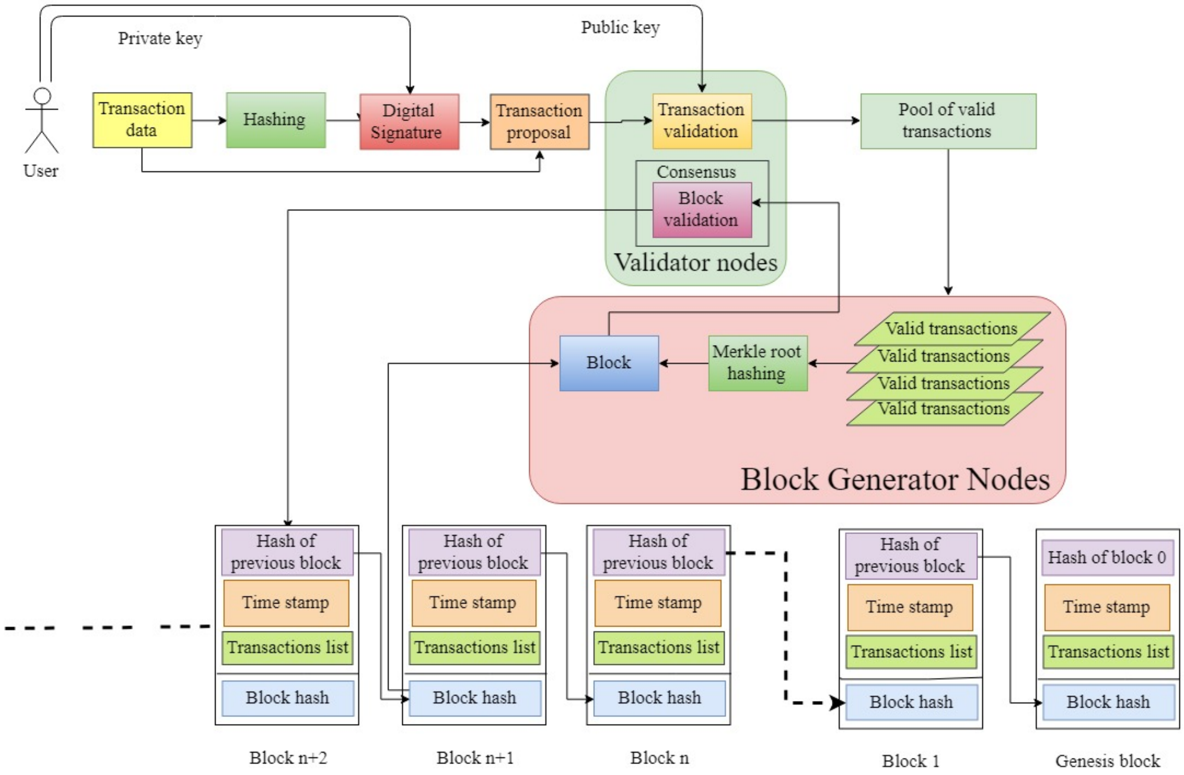

**Figure 2.** Overview of Transaction Execution Flow in Blockchain.

participants' nodes. This eliminates the need of an intermediary to process the transactions and maintain a ledger of the transactions records [1].

- Immutability: The transactions in the blockchain are packed in blocks. Each block in the chain is linked to the previous block using a cryptographic hash function. Thus, any attempt to modify the content of a block will affect the subsequent blocks in the chain. Consequently, a malicious attacker requires to computationally change all the succeeding blocks in the chain to modify a particular block. This becomes difficult as the copy of the chained blocks is replicated over multiple nodes.
- Transparency: The transactions records are in stored in a ledger, whose copy is replicated on multiple network nodes. The ledger is only updated when the majority of the nodes in the network reach a consensus about the transaction's validity. Changes to the network are publicly visible making the network more transparent and secure.
- Traceability: The distributed, transparent nature of the blockchain technology makes it easier to trace any transaction event, such as in supply chain [25]. Each update in the state of the asset can be traced down back to its origin. This helps in making the network more secure, efficient and transparent.
- Trustless: Blockchain allows transaction of assets between unknown parties who do not trust each other. By distributing the ledger across several nodes in the network and updating this ledger via a consensus ensures the validity of transactions in an untrusted environment.

### *3.2. Transaction Execution Overview*

Figure 2 shows the transaction execution flow in a blockchain network. It uses the following components:

- Transaction: A process that changes the state of the blockchain ledger. Depending on the application, the transaction can be the transfer of a financial value or the execution of a smart contract.

- Block: It consists of a block header and a block data. The header consists of the block's metadata information such as the merkle tree root hash, the previous block's hash, the time stamp, and the block version, whereas the data consists of a set of valid transactions [1].
- Merkle tree root hash: All the transactions in the block are hashed individually using a hashing algorithm. The hash values are then combined pairwise and are hashed again until a single hash value is obtained. This value is known as the merkle tree root hash value.
- Block hash: It is the unique identifier of a particular block and is obtained by hashing the block header twice [26].
- Previous block' hash: It is the hash of the block preceding the current block in the chain. The preceding block is known as the parent of the current block. The use of previous block's hash value in a block header is to ensure the immutability of blockchain ledger.
- Time stamp: It indicates the time at which the block is created.
- Block version: It indicates the version of the blockchain protocols used.
- Mining: It is the process of adding the valid transactions in a block and broadcasting that block to the network.
- Genesis Block: This is the first block in the ledger. All the following blocks in the chain are linked to the genesis block. The genesis block generally includes the configuration for the network characteristics, the consensus protocol to be used, the access control rights, the hash function, the block generation interval, and the block size.

The execution flow consists of the following steps:

1. Transaction proposal: the user first hashes the transaction data using a hash function [27] for later verification of data integrity. The hashed data is then encrypted using the user's private key to provide user authentication and the encrypted output is known as the digital signature of that transaction. The transaction data and the signature are broadcasted to the network.
2. Transaction and block validation: each full node in the network validates the transaction by performing two tasks: a) user authentication by decrypting the digital signature using the public key of the proposing user, and b) data integrity by hashing the transaction data and comparing it with the decrypted signature. The valid transaction is broadcasted to the block generators (miners) in the network. A selected miner (based on consensus) verifies the valid transactions and group them in a block in a way that the block size does not exceeds a predetermined threshold. The miner computes the merkle root hash value. The summarized hash of all the transactions by the merkle root provides an efficient process to verify a transaction in a block. To verify whether a transaction is included in a block or not, a node just requires the hash values of the merkle path connecting the transaction to the merkle root. Consequently, a node that do not maintain the entire copy of the ledger can verify a transaction by requiring the path without the need to receive the entire block, reducing communication overhead. To verify a transaction in a block consisting of $n$ transactions, a node requires only $\log_2 n$ hash values using merkle root as compared to $n$ hash values when not providing the merkle root hash value in the block header [26]. After calculating the merkle root hash value, the block's hash is generated. The miner broadcasts the block to the network. The validating nodes verifies the validity of the block by checking the correctness of of the block's hash, its time stamp is greater than that of previous block, its height and size values, the previous block's hash value, and that all the transactions in the block are valid. Each validating node will append the valid block to their copy of the ledger.

The replication of the ledger in a blockchain eliminates the issues of network dominance and data stewardship by a centralized service provider in addition to the problems of a single point of failure and a high network latency. Ideally, the copy of the ledger should be consistent between the different nodes and should be highly available. However, in a distributed system where a network partition may happen, data messages could be delayed or lost. Therefore ensuring high consistency and high availability at the same time is a difficult problem. Consequently, a trade-off [28] should be achieved.

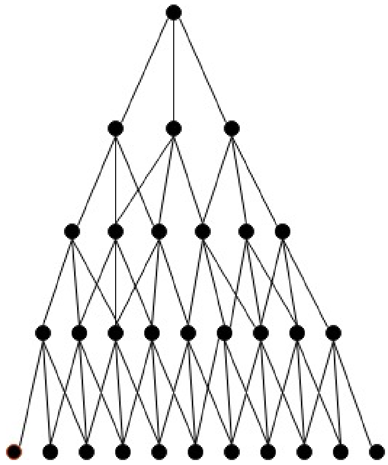

**Figure 3.** Hierarchy-based Connection of Nodes in the Blockchain Network.

The replication strategy used by the blockchain network is the monotonic prefix consistency (MPC) [29]. The transactions and blocks are broadcasted using the gossip protocol as shown in Figure 3. Each node in the network is connected to 'n' other nodes and each of them is connected to 'n' others and so on forming a hierarchy of nodes.

The broadcasting of transactions and blocks involves a high number of messages communicated over the network. In order to avoid communicating data to nodes which have already received it from some other node, the transaction and block data is not broadcasted directly to the nodes [30]. Instead, the node receiving a transaction or block first validates it and then sends an invitation message to its peers announcing the availability of data. The invitation message contains the hash of the transaction or the block. A node which receive an invitation message and does not have the transaction or block will reply back with a get data message [30].

### *3.3. Blockchain Organizational Framework*

Blockchain is emerging as a solution for distributed applications in a large-scale collaborative ecosystem for its secure and immutable characteristics, eliminating the need and the cost of intermediaries. The main goal is to build customers' trust in the network which helps a growing business for enterprises and individuals. Consequently, over the last decade there have been a prfoliferation of platforms and consensus protocols to develop applications in different domains. This makes it difficult for an applicaiton developer to chose a suitable platform. In addition, those platforms architectures and protocols have often misaligned with the goals of blockchain in building scalable solutions and the need for a green ecosystem. Figure 4 shows the classifications of the blockchain architectures and consensus protocols arranged in an organizational framework . The architecture taxonomy is based on the temporal evolution of the implemented platforms in terms of building blocks. The consensus protocols classification is based on computing power, non-computing capabilities, and voting algorithms.

The technology has evolved over time due to its adoption by different type of application domains, such as healthcare, education, logistics, governance, and robotics. This evolution is classified into 4 tiers as shown in Figure 5. The initial implementation of the blockchain in 2008 for the transfer of cryptocurrencies in a public network is known as Tier 1.0 [31]. Later in 2013, Tier 2.0 was introduced to facilitate digital transfer of non-financial assets using smart contracts [32]. Smart contracts are similar to paper contracts defining the rules and penalties related to an agreement. The use of public network in Tiers 1.0 and 2.0 has privacy concerns due to clear transactions in the ledger, and does not suit private applications. Consequently, Tier 3.0 came up in 2015 for the development of applications in a private network. However, due to the need of interoperability in a rapidly evolving collaborative ecosystem, Tier 4.0 was initiated in 2018 to solve interoperability issues [33].

doi:10.20944/preprints201908.0311.v1

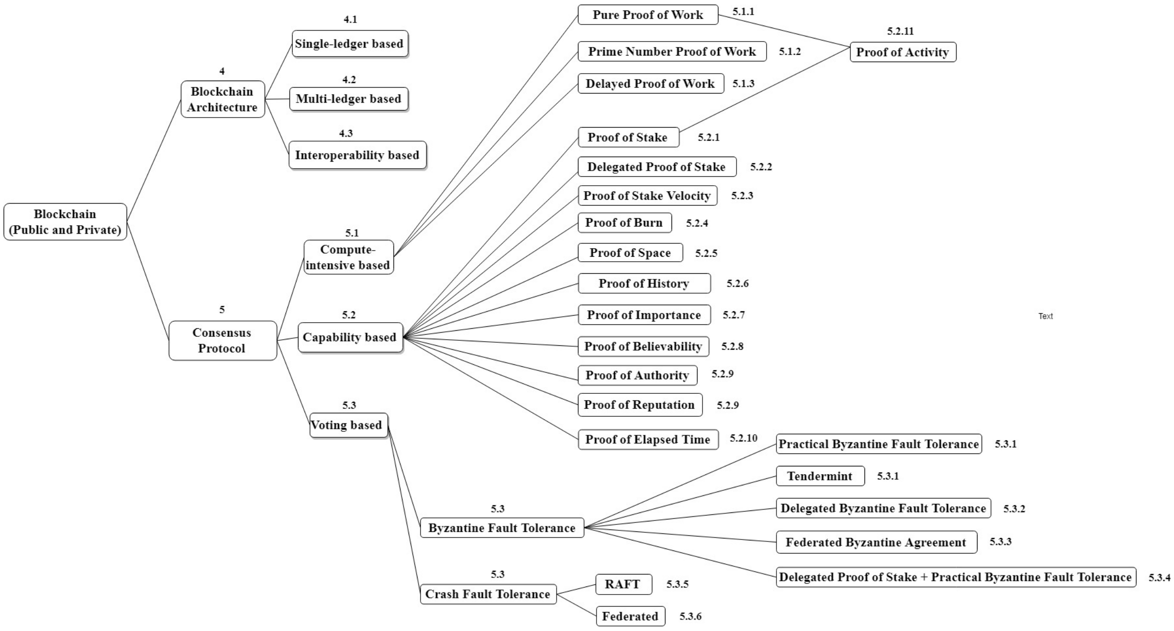

**Figure 4.** A Taxonomy-based Overview of Blockchain Literature Surveyed in this Paper. The numbers indicate the section/subsection in this paper.

**Table 1.** Comparison between public and private blockchain networks.

| | Public | Private |
|---|---|---|
| Network Join Permission | Open | Restricted/Authorized |
| Transaction Visibility | All members | Selected authorized members |
| Participants of Consensus Process | All block generators | Selected nodes |
| Trust in the Network | Not Required | Required |
| Data Privacy | Low | High |

A blockchain architecture is either public-centric or private-centric. Public blockchain, often referred to as permission-less blockchain, allows anyone to join the blockchain network without permission [1]. The user can join the network as a simple node, a validating node or a mining/block generating node. This type of network typically offers incentive for the users to participate in the consensus to encourage more participants to join the network. The identity of a network participant is kept pseudo-anonymous [34] using a public key where the participant's real identity is kept unknown and recognized by a pseudo name. However, the transaction data is kept public over the network leading to the issue of data privacy [35]. Private blockchain, also called permissioned, is an invite-only network from an authentication authority [1]. The network involves access control rights for ledger query and updates. Table 1 shows the comparison between the public and the private blockchain networks.

## 4. Evolution and Taxonomy of Blockchain Architectures

The proliferation of blockchain platforms introduced different architectures to satisfy the applications requirements in an evolving collaborative ecosystem. In this section, we present a temporal evolution of blockchain architectures. We provide retrospective analysis of these architectures and provide insights about the prevailing issues for future research directions. We derive a classification based on their characteristics and map them to the existing development platforms (Table 2).

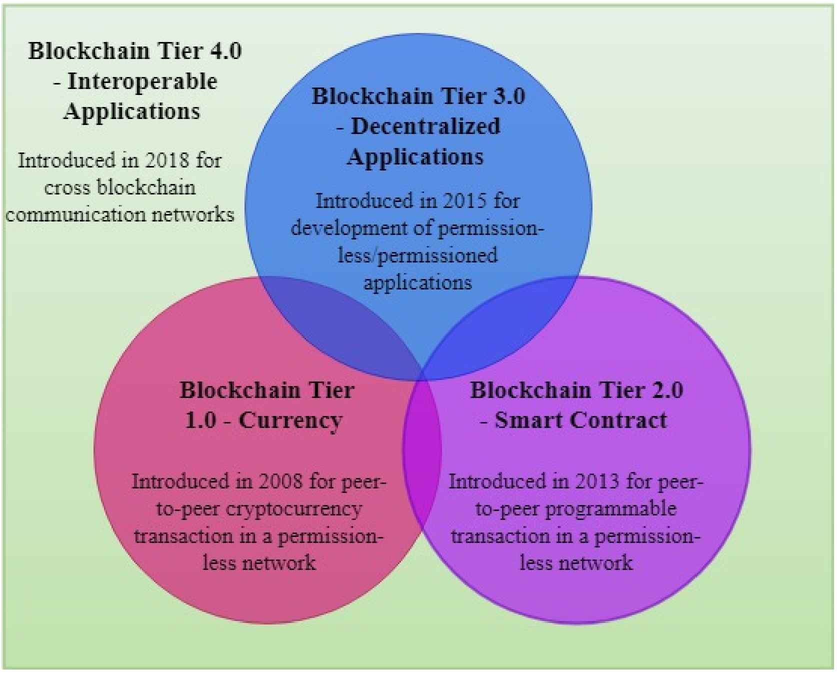
Blockchain Tier 4.0 - Interoperable Applications
Introduced in 2018 for cross blockchain communication networks
Blockchain Tier 3.0 - Decentralized Applications
Introduced in 2015 for development of permission-less/permissioned applications
Blockchain Tier 1.0 - Currency
Introduced in 2008 for peer-to-peer cryptocurrency transaction in a permission-less network
Blockchain Tier 2.0 - Smart Contract
Introduced in 2013 for peer-to-peer programmable transaction in a permission-less network

**Figure 5.** Blockchain Tiers.

### 4.1. *Single-ledger based Architecture*

Single-ledger based platforms were developed in tier 1.0 and then remained in tier 2.0 and 3.0. The corresponding architectures differ based on public, private, or hybrid networks applications.

#### 4.1.1. Single-ledger based Architecture for a Public Network

This architecture was introduced in 2013 by the Ethereum platform [36]. As shown in Figure 6 the network participants are represented by peers/nodes. A node can be simple, full, or mining. A client is the user issuing the transactions, using RPC to connect to the blockchain and an integration service to connect to an external system. An external systems is used if the validation of a transaction depends on external data such as the current weather, the price of a share market, or the currency exchange rate. However, if the external system is malicious, then the validity of the transaction becomes questionable [37]. The transaction execution flow in this architecture is as follows:

1. A client creates and hashes the transaction payload.
2. The digital signature of the hashed payload is generated.
3. The transaction payload and the digital signature are broadcasted to the network.
4. The transaction is validated by the validators and broadcasted to the miners.
5. A block of valid transactions is generated by a selected miner.
6. The block is broadcasted to the network.
7. The block is verified by the validators and the ledger is updated.

This architecture can be used applications domains, such as transportation, supply chain and project managements, digital content ownership, finance, and energy trading. However, it has limitations to implement private applications, such as healthcare, education, governance, and national policy because: 1) the architecture makes it is open to anyone to join the network and the transaction data is public, and 2) no access control mechanism is employed.

#### 4.1.2. Single-ledger based Architecture for a Private Network

A Single-ledger based architecture for a private network was introduced by adding building blocks to solve the issues of privacy and access control prevailing in the private network architecture.

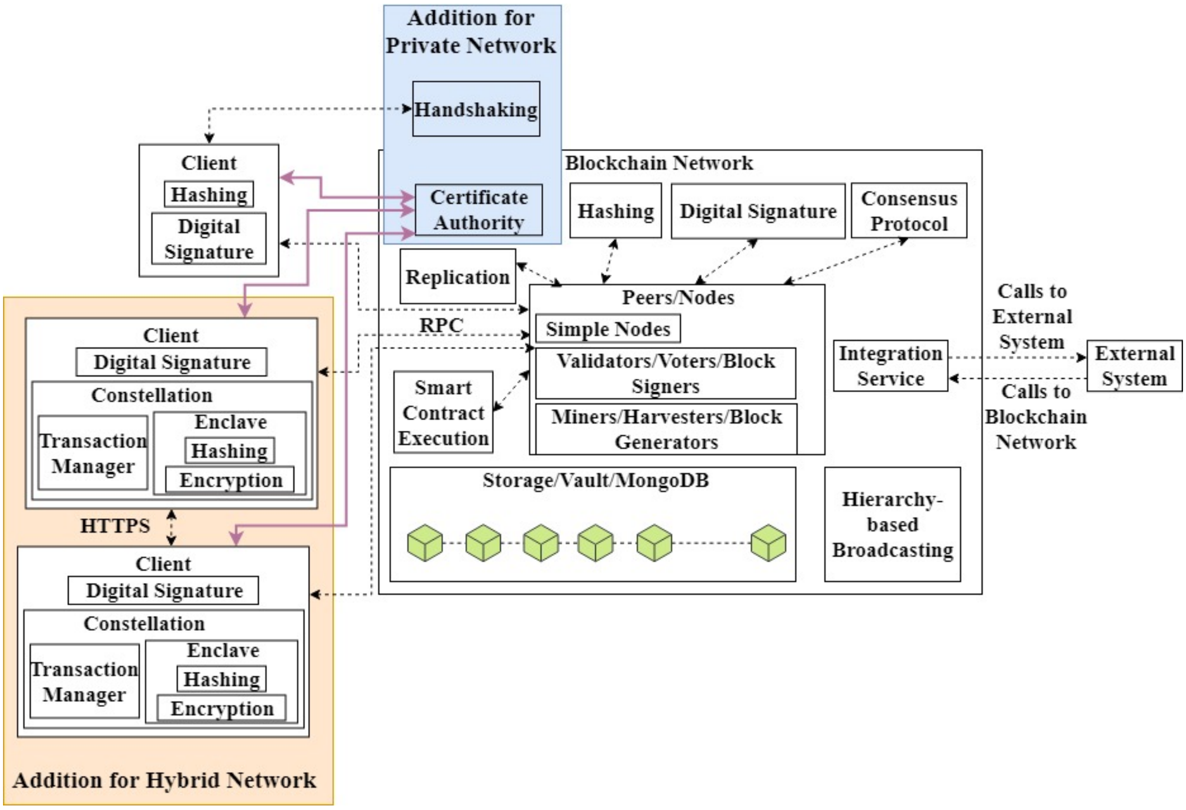

**Figure 6.** Single-ledger based Blockchain Architecture for Public, Private, and Hybrid Networks.

A certificate authority and a handshaking process are added as shown in Figure 6. The certificate authority provides authentication and authorization for users to join the blockchain network. The access control mechanism defines the ledger queries and updates roles for each participant in the network. The handshaking process ensures authenticity of the nodes participating in a transaction and establishes connections between those nodes. In 2015, the blockchain platform Multichain [38] introduced the process of handshaking as follows:

1. A transaction initiating node sends a challenge message to the other nodes participating in the transaction.
2. The message receiver nodes reply by signing the challenge message using their private keys.
3. The message sender node authenticates the signature by using the receivers' public keys.

The single-ledger based architecture for private network is used by various development platforms such as Hyperledger Burrow [39], Chain core [40], Hyperledger Sawtooth [41], Hydrachain [42], Hyperledger Iroha [43], Burst [44], NEM [45], and BigchainDB [46].

### 4.1.3. Single-ledger based Architecture for a Hybrid Network

To support the development of applications of hybrid nature (private transactions in a public ledger), blockchain platform Quorum [47] in 2016 introduced a constellation building block to the public architecture, as shown in Figure 6. Examples are real estate, social networking, retail industry, healthcare, and research.

A constellation allows the submission of transactions in a private way by using encryption. It includes a transaction manager and an enclave. The transaction manager keeps the transaction data private and secure by broadcasting the hashed encrypted data to the network. The hashing and encryption/decryption operations are performed by the enclave. The transaction execution flow for the hybrid network is as follows:

1. Participant A (sender) creates and sends the transaction payload to its transaction manager along with the public keys of all the participants involved in the transaction. Let us suppose that participants B and C are the receiver of the transaction.
2. The transaction manager of A stores the transaction payload at its local disk and sends the payload to its enclave for encryption.
3. Participant A's enclave encrypts the payload using a generated symmetric key and hashes the encrypted payload.
4. The enclave encrypts the symmetric key using the public key of A,B, and C individually.
5. The encrypted payload, its hash, and the encrypted symmetric key is sent to the transaction manager.
6. The transaction manager stores the encrypted payload and key using the hash value as an index. The encrypted payload, the encrypted key using B's public key, and the hash value are then transferred to the transaction manager of B securely via Hyper Text Transfer Protocol Secure (HTTPS) [48]. Similarly, the encrypted payload, the encrypted key using C's public key, and the hash value are then transferred to the transaction manager of C.
7. The transaction manager of B and C responds with an acknowledgment response to the transaction manager of A.
8. Participant A's transaction manager replaces the payload data on its local disk by the hash value and broadcasts it to the network.
9. Upon receiving the hash value, the network nodes search for the received hash value in their respective transaction manager. The nodes having the hash value pass the hash value, and the encrypted payload and key to their enclaves (B and C in this case).
10. The enclave decrypts the symmetric key using the participant's private key, and then decrypts the transaction payload using the symmetric key. The decrypted payload is then sent to the transaction manager.
11. The transaction manager of participants A, B, and C executes the transaction. The nodes which are not part of this transaction skip the broadcasted hash value.
12. The transaction is added to the ledger using its hash value.

In summary, single-ledger based blockchain architecture for public network can be used for application domains that do not require private transactions and authentication for the users to join the network, whereas the architecture for private network should be used to build a blockchain within an organization or a federation of organizations. The architecture for the hybrid network should be used for the public domain applications requiring confidential transactions between some network participants. As in a public and hybrid network, anyone can be a full node and/or a mining one, the number of messages transfers and blocks validations increases with the growing network limiting scalability. Comparing the number of encryption/decryption operations in a transaction flow, the private network has less operations than the public network thanks to its defined access control to the blockchain. Both public and private architectures store all the transactions data in clear in the ledger. However, the hybrid architecture has less encryption/decryption operations than the public and private because the nodes which are not involved in a private transaction store only the transaction's hash. The encryption/decryption operations are computationally complex and energy hungry [49].

### *4.2. Multi-ledger based Architecture for a Private Network*

In 2016, blockchain platform Hyperledger Fabric [50,51] introduced multi-ledger based architecture for private network, as shown in Figure 7. This is to enable confidential and private transactions between subgroup of participants within an organization or federation. The architecture divides the blockchain network into channels to enable private transactions between the members of a channel [52]. To perform a private transaction between participants within a subgroup, the architecture uses collection. This is because creating channel within a channel is a CPU intensive process consuming high energy as compared to that creation of a collection within a channel [53]. The validators are known as peers and the miners are known as orderers in this architecture. There are two types of peers:

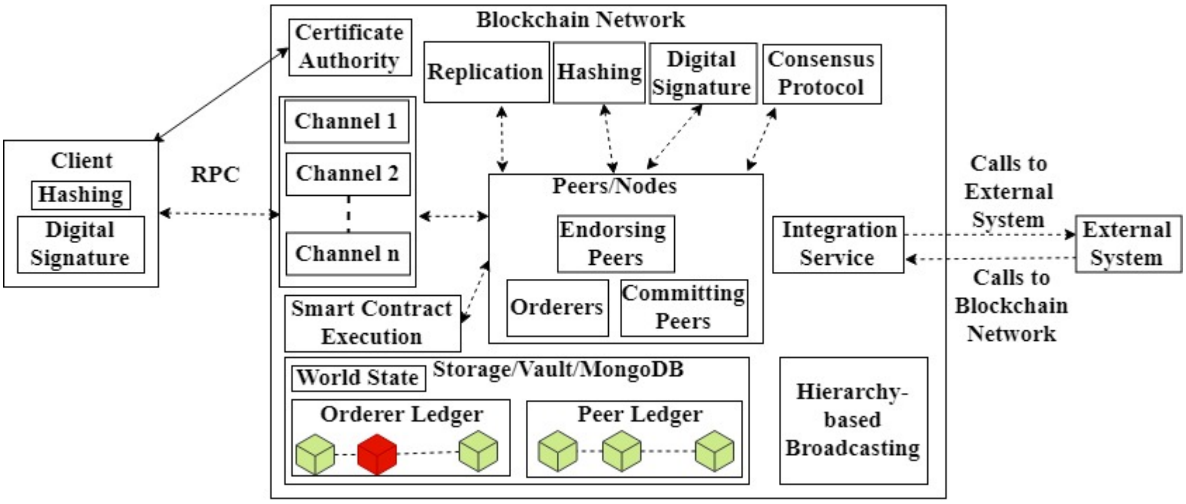

**Figure 7.** Multi-ledger based Blockchain Architecture.

endorsing peers and committing peers. The ledger in this architecture comprises of two components: the world state and the ledger. The world state describes the current state of the ledger. The transaction execution flow in this architecture is as follows:

1. A client creates the transaction payload containing a specific channel to be used for that transaction. The payload is hashed to generate a transaction ID used for future references.
2. The transaction data is digitally signed and broadcasted to the network. The broadcasted transaction payload is referred as transaction proposal.
3. The transaction payload is validated by the endorsers. Each endorser execute the chain code associated with a valid transaction and records the change in the world state. However, no change in the world state is applied at this point.
4. Each endorser digitally signs the transaction and sends back a proposal response to the client.
5. The client waits unless a predefined number of proposal responses are received. The client aggregates all the responses and sends it along with the transaction to the orderer.
6. The orderer generates a block of transactions (valid and invalid) received from different client and append the block to its ledger. The block is broadcasted to the peers in the network.
7. Each peer in the network verify the validity of the transaction in the block. The peer creates a block of valid transactions (i.e. transaction having threshold endorser signatures or the world state output is not correct) known as vblock. The vblock is the appended on to the chain of the peers. The invalid transactions are logged for future reference but are not included in the block.
8. The client is notified regarding the successful execution of the transaction.

The multi-ledger based architecture can be used in applications domains that involve several collaborating organizations requiring confidentiality of transactions among different subgroups of the network. For example, collaborating universities, banks or hospitals. The architecture is implemented by Oracle [54].

Compared to the single-ledger based hybrid architecture that also enables confidential transactions, the multi-ledger architecture requires more number of encryption/decryption operations. This is because in a hybrid network the encryption/decryption operations will be only performed by the nodes involved in a transaction, whereas in the multi-ledger architecture all the endorsing and committing peers will perform the decryption operation.

In Summary, applications domains where the transactions are big in size, such as in the healthcare, including laboratory results and images, the blockchain architectures suffer from the issue of throughput. This is because of the limited block size in the blockchain network which reduces the number of transactions in a block if the transactions are big. In that case, transactions can be compressed as introduced by the blockchain development platform Credits [55]. The transaction

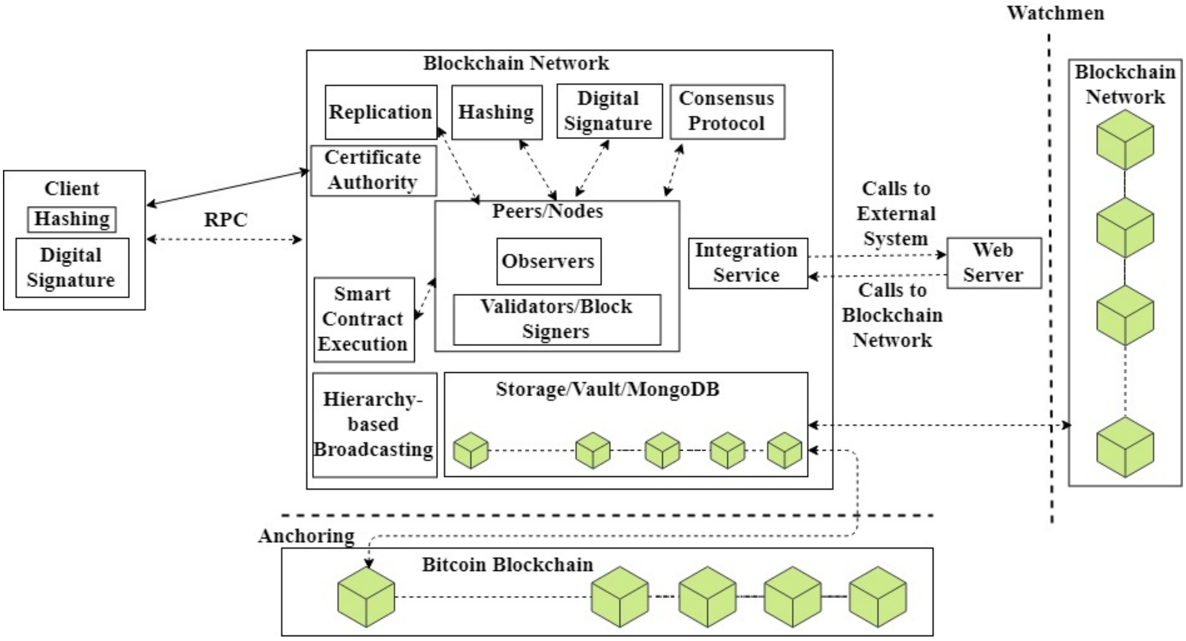

**Figure 8.** Interoperability based Blockchain Architecture.

are compressed using a lossless compress algorithm, called Deflate; a combination of LZ77 [56] and Huffman algorithms [57]. In addition, scalability has to be addressed in a blockchain architecture as a ledger is replicated among the majority network members. This increases the computational and communication overhead. As a research direction, scalability can be enhanced for instance by introducing a lightweight blockchain architecture that divides the network participants in clusters and selects a cluster head(s) from each cluster [58]. The cluster head(s) can be selected either based on voting or based on number of incident edges from a node [59]. The ledger can be then replicated only on the clusters heads and the clusters participants can query the ledger to get the transactions information.

### *4.3. Interoperability based Architecture*

The rapid adoption of blockchain by different applications domains motivated the development of many blockchain platforms. However, these platforms support different programming languages, smart contracts forms and structures, and communication protocols making it difficult to inter operate between different blockchains. In 2017, blockchain platform Elements [60] introduced the architecture for the interoperability between public and private networks blockchains, as show in Figure 8. The architecture is also used to enhance the security of a blockchain by linking it to another blockchain. The platform Openchain [61] uses it to link its blockchain (let us call it side-chain) to the bitcoin blockchain (let us call it main-chain). Whenever a block is added in the side-chain, a cumulative hash for that block is calculated by hashing the block's hash with the previous block's cumulative hash. The current cumulative hash is then recorded in the block of the main-chain. This process is known as anchoring and it ensures high immutability of the side-chain.

In order to perform a transaction, the architecture uses a service called Watchmen. Watchmen is an authorized member that enables secure and verified inter-chain assets transfer. The architecture is developed by Lisk platform [62]. The transaction execution flow in this architecture is as follows:

1. The transaction payload is created and signed by a node in the first blockchain, and sent to Watchmen.
2. Watchmen validates the transaction and releases it to the second blockchain network.
3. The block signer (similar to the miner) in the second chain adds this transaction in the block and the ledger is updated once the block is verified.

The interoperability based architecture is developed for a single-ledger based blockchain. However, it can be used for multi-ledger blockchain.

Table 2 shows the taxonomy of different blockchain platforms for the classified architecture types. It shows the specificity of each platform, the network type, the operating systems supported, the programming language(s) to develop applications. It also shows whether a platform supports or not applications' portability. The portability is achieved by generating bytecodes.

**Table 2.** Taxonomy of Different Blockchain Development Platforms.

| **Architecture Type** | **Supporting Platform** | **Platform Specificity** | **Network Type** | **Operating Systems Supported** | **Programming Languages** | **Applications Portability** | **Applications** |
|---|---|---|---|---|---|---|---|
| Single-ledger based | Ethereum [36] | - | Public | Linux, Windows and Mac OS | Solidity, Serpent, Lisp-Like-Language | Yes | Transportation, finance, supply chain, digital content and project managements, and energy |
| | Hyperledger Burrow [39] | - | Private | Linux | Go | Yes | Education, healthcare, governance, national policy, and research |
| | Chain core [40] | Validator is known as block signer and miner is known as block generator | | Linux, Windows and Mac OS | Ivy | | |
| | Hyperledger Sawtooth [41] | Validator is known as journal and the role of miner is performed by a validator | | Linux, Windows and Mac OS | Java, Python, JavaScript, Go, C++, Rust | | |
| | Hydrachain [42] | The role of miner is performed by a validator | | Ubuntu | Python | No | |
| | Hyperledger Iroha [43] | Validator is known as peer and miner is known as orderer | | Linux and Mac OS | Python, Java, JavaScript, C++ | | |
| | Burst [44] | - | | Linux, Windows and Mac OS | JavaScript | | |
| | NEM [45] | Miner is known as harvester | | Linux, Windows and Mac OS | Java | | |
| | BigchainDB [46] | Validator is known as mongodb server and the role of miner is performed by a validator | | Ubuntu | JavaScript, Python | | |
| | MultiChain [38] | The role of miner is performed by a validator | | Linux and Windows | Python, C++, JavaScript, Ruby, Go | | |
| | Quorum [63] | The role of validator is performed by the constellation component. | Hybrid | Red Hat 6.5/7, SUSE 11m3/12, Ubuntu | Python,Java | Yes | Social networking, healthcare, real estate, retail industry, and research |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | Credits [55] | Validator is known as trusted node and miner is known as main node | | Linux and Windows | Java | | |
| Multi-ledger based | Hyperledger Fabric [50] | Validator is known as peer, miner is known as orderer, and smart contract is known as chain code | Private | Ubuntu 14.04/16.04 and Mac OS 10.12 | Java, Go, Node.js | No | Education, healthcare, and governance |
| | Oracle [54] | | | Linux, Windows and Mac OS | Java | | |
| Interoperability based | Elements [60] | Miner is known as block signer and watchmen performs the role of validator and also supports interoperability | Private | Linux, Windows and Mac OS | C++ | No | Collaborative ecosystems such as smart cities |
| | Lisk [62] | - | | Ubuntu 14.04.5, Windows and Mac OS | JavaScript | | |
| | Openchain [61] | The role of miner is performed by validator and simple node is known as observer | | Linux, Windows and Mac OS | JavaScript | | |

## 5. Evolution of Consensus Protocols in Blockchain

Because a blockchain network is not controlled by any single authority, a transaction is verified after reaching a consensus by the majority of the network participants. However, there prevails a gap between the existing algorithms and the applications' needs in terms of scalability, complexity, cost effectiveness, and energy efficiency. In this section, we present a taxonomy of different consensus protocols used in the blockchain literature by classifying based on compute-intensive, non-computing capabilities, and voting. In particular, we provide a temporal evolution of these algorithms with a retrospective analysis to highlight the underlying issues. We also discuss some possible solutions to address these issues.

### *5.1. Compute-Intensive based Consensus Protocols*

Compute-intensive based consensus protocols are energy-hungry mining algorithms. Different compute-intensive consensus protocols used in the blockchain literature are discussed in this section.

### 5.1.1. Pure Proof of Work (PoW)

PoW was first introduced by Dwork et al. in the year 1992 to reduce the number of spam emails by making it computationally difficult to send multiple emails simultaneously [64]. This is by using a pricing function whose value has to be computed by the sender of the email. The calculation of the pricing function involves cryptography hash operations making it computationally complex and time consuming, thus preventing spam emails. Later in 2002, Back et al. proposed PoW using a hashcash function to decrease the number of spams emails. Hashcash uses the cryptographic hash function SHA-1 [65]. The message content, the time stamp, the recipient's email address, and a counter value are inputed to the hash function. The sender will use brute-force [66] method by incrementing the counter value, until a hash value with a predefined number of leading consecutive zeros is obtained. The hash and the counter values are then added to the email header. The recipient can easily verify the hash value by regenerating it [67]. For spammers sending thousands of random of emails every second, calculating the hash values will cost their time and computing resources and energy, discouraging them from doing the work.

The PoW algorithm was then used in the year 2008 by Nakamoto et al. for peer-to-peer transactions of cryptocurrencies (Bitcoin) without the involvement of any third party [2]. In PoW the mining nodes compete against each others to generate a block such that the block's hash should have a predefined number of consecutive leading zeros and below a threshold hash value. The number of leading zeros and the threshold determines the difficulty of the algorithm, and it is dynamically adjusted to maintain a constant block generation interval. The difficulty of the network increases exponentially with increasing number of zeros. In the bitcoin network, the difficulty is adjusted after every 2016 blocks to maintain the average block interval of 10 minutes [68]. Equations 1 and 2 show the calculation of the difficulty and the target value in the network respectively. The difficulty for the next 2016 blocks to be mined in the network $D_{new}$ is increased or decreased compared to the current difficulty level $D_{current}$ for the last mined 2016 blocks. The change is calculated based on the ratio of ideal time to generate 2016 blocks i.e. 20160 minutes (1 block every 10 minutes) and the actual time required to generate the last 2016 blocks. The target for the next 2016 blocks to be mined $Target_{new}$ is calculated similarly using the current target value $Target_{current}$ as the reciprocal of the difficulty.

$$D_{new} = D_{current} \times \frac{20160}{T_{2016}} \tag{1}$$

$$Target_{new} = Target_{current} \times \frac{T_{2016}}{20160} \tag{2}$$

To calculate a block's hash, the mining node will combine all the valid transactions in a block and calculate the merkle root hash value using SHA-256 hashing algorithm [69]. The merkle root hash value, the time stamp, the previous block's hash, the block version is then combined with a nonce value and inputted to the hash function. The nonce is a variable whose value is adjusted by brute-force method in order to produce the desired block hash. Once the nonce generating the desired hash is found, the miner adds the nonce value to the block header, and broadcasts the block to the network. All the other miners stop the mining process and check whether the proposed block is valid or not. Once valid, the block is appended to the chain and the miners start mining the next block. The miner whose block is appended to the chain receives a mining incentive for the computational power utilized. This incentive is divided into two parts: a transaction fee and a mining fee. A transaction fee is associated with each transaction in the block and its value is subject to the client. A mining fee is an incentive given to the miner by the entire network [70]. Miners compete in mining a block to win the incentives which exacerbates the energy consumption problem.

The bitcoin network implementing PoW currently has the throughput of 4-7 transactions per second. However, Gervais et al. developed framework to evaluate the performance of blockchain and showed that the throughput PoW-based blockchain can increase up to 60 transactions per second achieving the same level of security as in current PoW network [71]. PoW is widely used and provides a security service in the blockchain, as miners would refrain from mining invalid or malicious transactions, due to the invested computing power to mine a block. However, miners are now forming groups known as mining pools to solve the PoW puzzle. Each miner in a pool uses its computing capacity, and the mining reward is divided among the miners based on their mining contribution [72]. If a mining pool owns more than 50% of the network's computing power, then it is likely that those miners would be able to prevent the validation of proposed transactions, and consequently stop the transactions between users [73]. This is known as the problem of 51% attack in the PoW-based blockchain network. Moreover, PoW suffers from possible security attacks such as routing, sybil, eclipse, time jacking, and bribery attacks as discussed by Conti et al. [74]. Also, it favors the rich, as the chance of mining a block by a miner is proportional to the hardware computational resources owned by that miner and also requires specialized mining hardwares. In addition to the mentioned problems, there is the issue of high energy consumption by the hardware resources while brute-forcing the nonce value. In 2017, the bitcoin mining used around 30.14 TerraWatt hours (TWh) of energy, which is equivalent to energy usage of the entire Ireland in a year [3]. The annual energy consumption as of 22 June 2019 is 67.937 TWh [75]. This increasing amount of energy consumption is affecting the environment with increasing global warming. According to a research, the annual carbon dioxide emissions by the bitcoin network using the PoW are as high as 22.9 million metric tons, almost equivalent to the amount produced by the countries like Sri Lanka and Jordan [4]. The PoW consensus is implemented in Bitcoin, Litecoin [76], and Dogecoin [77] networks.

### 5.1.2. Prime Number Proof of Work (Prime number PoW)

In 2013, King S. proposed prime number proof of work to channelize the high amount of energy consumption of PoW for dual use [78]. Prime number PoW involves computationally complex calculation of Cunningham chain of prime numbers that can be used in cryptography systems. Cunningham prime chains of specific lengths has a cryptographic significance as they are used to implement auto-recoverable auto-certifiable cryptosystem enabling secure, robust, and recoverable file system [79]. Consequently, the high amount of energy consumption serves for the security of the blockchain network as well as for development of cryptography methods. The computed prime chains are published in the ledger. The chain of prime numbers must satisfy two requirements: 1) it should be a Cunningham chain [80], and 2) its length should be larger than a target length. The prime number PoW accepts one of three prime chains types: 1) Cunningham chain of first kind [81], 2) Cunningham chain of second kind [81], and 3) bi-twin chain [82].

To ensure that a prime chain is caluclated for each individual block, the calculation is made block specific. This is by requiring the first element of the chain to be divisible by the hash of the block header. The quotient of the division then becomes the proof of work certificate. The certificate is then hashed along with the block header to calculate the block hash. The validitiy of the block is verified by checking whether the hash of the block is correct, the prime chain length is above a target length and the prime chain is valid. In order to check the primality of the prime chain, the miners use the classical fermat test [83] together with Euler-Lagrange-Lifchitz test [84]. If the block is valid, then it is appended to the chain and the miner receives incentive which includes the mining fee and the transactions' fee.

To have a constant block generation interval, the chain length is set as the difficulty in the prime number PoW. For a prime chain $p_1$, $p_2$, ..., $p_k$ of length k, the difficulty for the next prime chain is then calculated using equation 3.

$$d = k + \frac{p_k - r}{p_k} \tag{3}$$

where, r is the Fermat test remainder of the number $p_k$.

The prime number PoW consumes a lot of energy due to compute-intensive competitive calculation of the mining proof. Furthermore, the block verification time in prime number PoW is more than that in PoW [78]. This is because, in PoW the mining proof is verified by performing only one hash operation (i.e generating the block hash by using the provided nonce value), whereas in the prime number PoW, the mining proof is verified by performing two hash operations (one for the calculation of mining certificate and one for the block hash) and two primality tests. In addition, this consensus protocol has not been tested yet for its performance and protection against security threats. The prime number PoW is only used by the cryptocurrency primecoin [85].

### 5.1.3. Delayed Proof of Work (DPoW)

The PoW is an energy hungry secure consensus protocol used by the blockchain network. Delayed Proof of Work (DPoW) proposes to use this compute-intensive security of PoW to secure other blockchain networks that uses an energy-efficient consensus protocol. Therefore, DPoW is a hybrid consensus method allowing a blockchain to secure itself using the mining power of a PoW blockchain. In DPoW, a group of notary nodes (elected by the stakeholders in the network) are responsible for generating a block. Each of the 64 notary nodes validate the transactions and create a block in a round-robin fashion without involving compute-intensive and energy hungry calculation of the mining proof. However, to ensure security of the network, the hash of the last created block in the DPoW blockchain is added to the PoW blockchain whenever a block is created in the latter. The block hash in DPoW is signed by 33 (52%) of the notary nodes before sending it to the PoW blockchain network. DPoW is implemented by the platform Komodo [86] which uses the bitcoin blockchain to recover from invalid transactions. DPoW solving the issue of high energy consumption of PoW and prime number PoW compromises network security when the communication between the DPoW and

doi:10.20944/preprints201908.0311.v1

the PoW blockchain is interrupted or lost. Moreover, the consensus protocol has not been tested yet for its performance and protection against security threats.

In compute-intensive protocols where the miners compete to mine a new block, it can occur that two miners separated geographically mines a valid block simultaneously and broadcasts it to the network. Depending on the location and network connectivity it may happen that part of the network nodes receives a block from one miner (let us call it as block A) and the other part receives a block from another miner (let us call it as block B). The nodes receiving block A verify the validity of the block and append it to their chain and the nodes receiving block B will append it on to their chain after verification. Now, when the nodes having block A in their chain receives block B, the nodes append the block B in a secondary chain as both the blocks A and B have the same parent block. Similarly, the nodes having block B in the main chain appends block A in the secondary chain once received. This problem in the blockchain when the chain divides into two parts is known as forking [26]. In order to resolve forking, the blockchain network uses the rule of the longest chain. This rule states that if the next block mined in the network will have block A as a parent block, then the blockchain with block A in the main chain will be considered valid as it becomes long compared to one with block B in main chain. Similarly, if the new block contains block B as the parent block, the blockchain with block B in the main chain will be considered valid. The blocks in the side chain are then removed and the chain continues to develop on the main chain. Generally, the blockchain fork is resolved within one block [26]. Therefore, to handle transactions in the forked block, all the transactions in a block in the blockchain are executed after certain number of blocks are appended to the longest chain. For instance, in bitcoin network, it takes 6 blocks confirmation for a transaction to get executed [87].

### 5.2. *Capability based Consensus Protocols*

The high energy consumption of networks using a compute-intensive based consensus protocol is due to its competitive approach. All the miners use their computing power to win the right to mine the next block in the network. Consequently, several consensus protocols in the literature were proposed to select a miner based on non-computing capability. The capability of a miner can be calculated based on various factors such as the amount of cryptocurrency owned by that miner, the contribution of the miner to the community, the trust the network has on the miner, or the amount of storage owned by the miner. In this paper, we classify those consensus under capability based protocols. Different capability based protocols are explained below.

#### 5.2.1. Proof of Stake (PoS)

To eliminate the competitive approach of PoW consensus consuming a high amount of energy, an alternate consensus known as Proof of Stake (PoS) was proposed in 2011 [88] which was later used by the crytpocurrency Peercoin (also known as PPcoin) in 2012 [89]. The miners in PoS are called forgers and the mining process is referred to as forging. At the beginning of a forging round, each forger deposits a certain amount of owned cryptocurrency coins in the network as stake, which is used by the protocol to select the next forger in the network. There are two forger selection methods in PoS: 1) coin-age selection based on the number of the days the coins are held at stake, and 2) randomized block selection based on the calculation of a hit value using the forger's private key.

In the coin-age selection method [89], a forger having the maximum value of coin age is selected to forge the block. Coin-age is calculated by multiplying the total number of coins that are being staked by a forger and the total number of days the stake is held as stated in equation 4. For example, 30 coins hold for 10 days will have coin age of 300 coin days. In order to participate in the process of forging, the coins must be staked for minimum of 30 days. The stake holding duration is involved in order to avoid repetitive selection of a forger having more number of coins and to make the process semi-random. However, it may occur that a malicious user increases its probability of forging a block by holding the stake for a long period of time. To prevent this situation the stake holding period is capped at the maximum of 90 days by the protocol. Once a block is created by a forger, the coin-age value of the coins staked by that forger becomes zero.

$$\text{Coin age} = \text{Total coins staked} \times \text{Number of days those coins are held at stake} \tag{4}$$

In the randomized block selection method [90], a forger having a specific hit value is selected for forging the next block. In order to calculate the hit value, each forger encrypts the hash of the previous block using its private key. The encrypted value is hashed and the first 8-bytes of the hashed output are converted into a number known as hit value. The use of private key in the calculation generates a unique hit value for each forger in the network. The forger having the hit value below a target value is selected for the process of forging. The target value is calculated using equation 5. To make the selection based on the capability of the miner, the calculation of the target value involves the amount of coins staked by the miner. Consequently, the target value of each forger in the network is different and the value is higher for a forger having more coins at stake, increasing the chances of that miner to forge the next block. Moreover, to make the target value non-deterministic, the calculation involves the time elapsed from the last block forged changing the target value every second.

$$T = T_b \times S \times B_e \tag{5}$$

where $T_b$ is the base target value calculated by multiplying the previous block's target value and the amount to time that was required to forge that block, S is the time elapsed since the last block forged and $B_e$ is the coins at stake.

If the hit value of more than one forger is below the target value, then the forger having high value of cumulative difficulty is selected. The cumulative difficulty mentioned is calculated as stated in equation 6 [90]. The forger who forges the block receives the transactions' fees of all the transactions in the block. There is no mining fee in PoS. If the forger tries to generate a malicious attack, the coins at stake are lost discouraging the forgers to behave as a bad actor.

$$D_{cb} = D_{pb} + \frac{2^{64}}{T_b} \tag{6}$$

where $D_{pb}$ is the previous block's difficulty.

Compared to PoW where the miners can form a mining pool to have 51% of the network capacity, in PoS it is difficult for a group of forgers to accumulate 51% of the cryptocurrency. This is because of the increasing price of digital currency (required in PoS) as compared to the computing hardware capacity (required in PoW). Consequently, PoS is more secure towards 51% attack than PoW. However, PoS favors the rich by increasing their chances to forge the next block as the selection of the forger depends on the amount of cryptocurrency owned by a forger. In addition, the consensus protocol reduces the transaction flow in the network as the forgers are lured to keep the cryptocurrency at stake discouraging financial transactions. Moreover, the protocol does not prevent malicious users from generating invalid blocks, as the staked coins as returned back to them as compared to PoW where the computational power used by the miner is non retrievable. Also, this consensus protocol has not been tested yet for its performance.

### 5.2.2. Delegated Proof of Stake (DPoS)

Delegated proof of stake (DPoS) was proposed by Larimer in 2014 to solve the issue of rich getting richer in the PoS by selecting the forgers based on election rather than on the amount of staked coins owned [91]. In DPoS a group of nodes known as witnesses (also referred to as delegates) are elected based on a voting process, where each network node owning cryptocurrency participates in the process of voting. The weight of a node's vote is proportional to the number of coins that a node owns. A node is allowed to vote multiple witnesses with a single vote for each witness. The first N witnesses having the highest votes are then selected for the mining process in order to avoid a single witness from

doi:10.20944/preprints201908.0311.v1

mining all the blocks. The number of witnesses (N) is selected such that 50% of the nodes have voted for these many witnesses. For example, if majority of the nodes have voted for atleast 10 witnesses, then the first 10 witnesses having the highest number of nodes will be selected. Each witness in the group mines a block in a round robin fashion. The list of witnesses is changed after a fixed period of time. If a witness fails to produce a block in a given time slot, the next witness is selected from the group. Once all the witnesses in the group have had their turn, the list of witnesses is shuffled and the round-robin continues. The shuffling aims to prevent a deterministic approach, vulnerable to attack, in which the next miner is known in advance. DPoS consensus algorithm is used by the cryptocurrency trading platforms Bitshares [92], Nano [93], and Cardano [94]. The protocol does not consider the case where each node votes for itself. It has not been tested yet for its performance and protection against security threats and does not solve the issue of reduced transaction flow of PoS.

#### 5.2.3. Proof of Stake Velocity (PoSV)

To address the economic issue in PoS where a node may not perform transactions in order to increase its chance of being selected as the next forger, Proof of Stake Velocity (PoSV) was proposed by Ren in 2014 [95] to prevent the financial flow in the system, which is necessary for an economy to grow. This is done by using an exponential growing function for the coin age calculation as compared to linear function used by PoS (Equation 4). Consequently, the growth rate of the coin age follows an exponential decaying line [95]. The exponential decay constant is selected in a way that the coin age reduces to half at every fixed interval of time, which we define as half-time. For instance, if the half-time is 10 days, then each coin will have coin age of one coin day per calendar day for 10 days. During the next 10 days, each coin will have coin age of half coin day per calendar day. Similarly for the next 10 days, it will be one fourth coin day per calendar day. As the holding period of the coin approaches to infinity, the coin age asymptotically approaches to 2 coin months [95]. Due to the exponential decay in the growth rate of coin age, the newly accumulated coins will dominate the stale coins encouraging the stake holders to actively move their stake by transacting with counter parties.

PoSV is an attempt to encourage financial flow in the network. However, if the counter parties exchange cryptocurrency with each other just for the purpose of reinitialization the coin age, then the economy will not get benefit from this financial flow. Moreover, the protocol still favors the rich and has not been tested yet for its performance and protection against security threats. PoSV is used by the cryptocurrency Reddcoin [96].

#### 5.2.4. Proof of Burn (PoB)

To address the issue of high energy consumption in PoW and the problem of retrievable staked coins encouraging malicious users in PoS, Ian Stewart proposed Proof of Burn (PoB) in 2014 [97]. In PoB, the miners need to burn the coins by sending them to a irretrievable address, known as eater address. The eater address has a public key associated with no private key making it impossible to retrieve the coins from that account. The coins once sent to this address are removed from the network and can not be further used. This discourages the malicious miners from mining an invalid block as a miner will spend coins in order to mine a block. The basic idea behind PoB is similar to PoW in a way that the miners invest in mining computing resources in order to increase their probability of mining the next block and in PoB the miners burn more coins which is analogous to buying virtual mining rigs. Consequently, a miner purchases the right to mine a block in PoB similar to purchasing computing resources for mining in PoW. In order to remove the dominance of the early adopters, the value of the burned coins decay exponentially with time. The transactions performed for sending coins to the eaters address are recorded separately from the other transactions taking place in the network. Once the transactions are recorded, a burn hash for each transaction is calculated using SHA-256, and the miner with the least value of burn hash wins the mining right. The burn hash is calculated using equation 7 [98].

$$Burn\ hash = (Internal\ hash) \times Multiplier \tag{7}$$

The internal hash is calculated by hashing together the burnt transaction's hash value, the time elapsed after burning the coins and the current block number. The multiplier is inversely proportional to the burned coins, increasing the probability of a miner burning more coins to be selected. However, to encourage continous participation of the miners, the value of multiplier increases exponentially lowering the probability of a miner to win with time. The value of multiplier is calculated using equation 8 [97].

$$Multiplier = \frac{e^{\frac{T_b}{T_d}}}{Burned\ coins} \tag{8}$$

where $T_b$ is the time elapsed from the time the coins were burnt and $T_d$ is the time after which the coin will decay.

PoB is used by cryptocurrency slimcoin [99] and third generation coin [100]. PoB favors the rich and has not been tested for its performance.

#### 5.2.5. Proof of Space (PoSpace)

To address the issue of rich getting richer in the previosuly discussed protocols in addition to the issue of high energy consumption in the computation-based protcols, Dziembowski et al. proposed proof of space (PoSpace) also known as proof of capacity in 2015 where a miner having enough disk space wins the right to generate the next block in the chain [101]. PoSpace is a two-step process: 1) plotting and 2) mining. The plotting step is a one time process, in which the hard disk of the miner is plotted using hash values to ensure the storage space dedicated by the miner. The plotting uses Shabal 256 [102] hash function that generates a 32-byte hash output value. The plotting begins by generating a 16-byte seed value containing the 8-byte account ID of the miner and 8-byte nonce number. The use of account ID makes the plot for each miner different. The initial nonce value is kept 0 and then incremented by 1 at each iteration up to $2^{64}$. Corresponding to each nonce value, a 256 Kibibyte (KiB) of hash value is generated by iterating the hash function 8192 times. This is done by initially feeding the seed value to the hash function to get the first hash value. The first hash value obtained is referred as hash #8191. The hash #8191 is then appended to the first seed value to form a new seed value and is again fed to the hash function to generate hash #8190. Next, this hash value is appended to the previous seed value and hashed again to generate hash #8189. This process is repeated for 128 iterations, after which the seed value becomes more than 4096 bytes. Thus, for all the remaining iterations from 129 to 8912, the last generated 4096 bytes is used as the seed value to generate the hash. This is to avoid computational overhead. Once all the 8192 hashes are generated, a final hash value is generated by hashing all the 8192 hash values and the first 16-byte seed value together. The final hash value is then XORed with all the individual hash values, and the results are stored in pairs. Each pair of hash values is termed as scoop, and for 8192 hashes there exists 4096 scoops. This process is repeated for all the nonce values between 0 - $2^{64}$.

The mining step is performed each time a new block is to be mined. In the mining step, a generation hash is calculated by the network which depends on the previous block in the chain. The generation hash is calculated by hashing the generation signature and the current block height together, where the generation signature is calculated by hashing the account ID of the previous block generator and the hash of the previous block. After calculating the generation hash, a scoop number is calculated by performing generation hash modulo 4096 (total number of scoops). From the hashed plot obtained in the plotting step, the 64 byte scoop data corresponding to the calculated scoop number for each nonce value is hashed along with the generation signature to generate a target value. The target value is divided by the base target value and the first 8-byte of the result is considered as a deadline value. The base target value is calculated using the block generation time of last 24 blocks. The minimum deadline value from the ones calculated for each nonce value is submitted by the miner along with the corresponding nonce value, and the account ID to the network. The network verifies the deadline by recalculating the scoop for that particular hash, and waits until the deadline time has passed. If no other node publishes a shorter deadline time than the one submitted, the node is selected for the generation of next block. The miner in PoSpace is known as forger and is rewarded with mining fee in addition to the transactions fee.

The main advantage of PoSpace is that it consumes less energy than compute-intensive based protocols and does not favor the rich as in the previosuly discussed capability based protocols. Moreover, the protocol does not require any specialised hardware for mining. However, PoSpace is hardly tested for its performance and can be prone to malware attacks as the plot of hashes stored in the hard disk can be easily attacked and tampered with. Moreover, the miner does not burn any energy or coins in order to mine the block, encourgaing malicious users to generate invalid block. PoSpace is currently used by cryptocurrency such as Spacecoin [103], Chia [104], and Burstcoin [105].

#### 5.2.6. Proof of History (PoH)

Proof of History (PoH) proposed in 2017 by Yakovenko aims to address the issues prevailing in compute-intensive based and above discussed capability based protocols [106]. In PoH, the validators are referred as the verifiers and the mining nodes are referred as the leaders or PoH generators. PoH uses SHA-256 hashing algorithm that runs over itself continuously with the output being the next input. The leader runs the hash function for a random starting value and takes the output and pass it as the input for the same function again. The leader records the output of the function every time and the corresponding counter value indicating the iteration. When a transaction takes place in the network, the leader verifies and combines it with the current hash output. This combination is then used as the next input and the counter value, the transaction and the hash output are recorded in the ledger. In this way, the transaction is recorded to have happened in time before and after a particular counter value. The ledger state is then passed to the verifiers who then verify that the transaction is valid and recalculate the hash output for all the counter values. The generation of hash can not be parallelized on a mulit-core architecture as the output of the function can not be known before hand. On the other hand, the proof can be verified in parallel by the verifiers on a multi-core architecture making it less time consuming. Equation 9 and 10 show the time taken for the hash generation and verification respectively on the same multi-core machine [106].

$$T_{generation} = \frac{Total\ number\ of\ hashes}{Hashes\ per\ second\ for\ 1\ core} \tag{9}$$

$$T_{verification} = \frac{T_{generation}}{Number\ of\ cores} \tag{10}$$

In the process of verifying the proof generated by the leader, each verifier signs the proof and send it back to the network. If majority of signatures are received in a predefined interval of time then the ledger status is updated or else it is denoted as the PoH generator failure. A new leader from the group verifiers is elected in the case of generator failure based on the amount of coins staked by each verifier.

As PoH do not require intensive and time consuming mining such as in PoW, it consumes less energy. However, PoH favors the rich for the selection of the leader making the process deterministic and centralized. Moreover, it is necessary to have a multi-core CPU architecture (although common now-a-days) in order to increase the speed of verification. In addition, as the hashing function is continously repeated even if there are no transactions, the ledger occupies more space as compared to the ledger generated by the other consensus protocols.

### 5.2.7. Proof of Importance (PoI)

Proof of Importance introduced in 2018 by the cryptocurrency platform NEM [45] addresses the issue of reduced transaction flow exisitng in the PoS protocol where the miners do not perform transactions in order to increase their chances of mining. In PoI, the miners are referred as harvesters and the process of mining is known as harvesting. A miner having the highest value of importance score in the network is selected to mine the next block. The importance score of a miner is calculated based on three factors: 1) the amount of crypto tokens vested by a miner, 2) the participants with whom the miner perform transactions, and 3) number and size of transactions performed by a miner. In order to be eligible for the mining process, a miner needs to have a minimum threshold amount of vested tokens, which we call vesting amount. 10% of the vesting amount vests each day [107], making it necessary for the miners to hold the tokens for a particular number of days. The higher the number of vested coins, the higher will be the miner's importance score. POI rewards miners in terms of importance score upon performing transactions with the nodes who has vested tokens. To avoid the reduction in transaction flow and holding of cryptocurrency by the miner, the protocol takes into consideration net transactions over time avoiding the miners to take the advantage by transacting back and forth [107]. Finally, the importance score also depends on the number of transactions performed by the miner in last 30 days with each transaction size being higher than a threshold value. The weight of a transaction $k$ that involved amount $x$ between $miner_a$ and $miner_b$ is calculated using equation 11 [45]. The miners in PoI receives the transaction fees as the reward.

$$w_{abk} = x.e^{ln(0.9)\,\frac{h-h_{abk}}{1440}} \tag{11}$$

where h is the current blockchain height and $h_{abk}$ is the height at which transaction k occured. The exponential decay function in the equation makes the weight of the transaction 0 after 30 days. The summation of all the weights for all the transactions between a and b is then taken and the net transaction value is calculated by taking the difference between $w_{abk}$ and $w_{bak}$ [45].

PoI discourages malicious users from mining invalid blocks compared to PoS and PoSpace as the miner is selected based on the recent transactions and the transacting parties. However, if the group of malicious attackers performs transactions amongst themselves, then the network security might be compromised. In addition, PoI implicitly favors the rich as the calculation of the importance score is based on the number of vested tokens, and the number and size of recent transactions. Moreover, this consensus protocol has not been tested yet for its performance and protection against security threats.

### 5.2.8. Proof of Believability (PoBelievability)

Proof of believability was proposed in 2017 to address issues of rich getting richer prevailing in PoS [108]. In PoBelievability the role of a miner is performed by a validator, where the validator with the highest believable score is selected for the generation of a block. The believability score is calculated based on the number of crypto token held by the validator, the number of previously validated transactions by that validator, and the number of servi [108] tokens earned by the validator. The servi token is a reward given to a validator for voluntary work to help the network. The voluntary work include providing storage space, dedicating computing resource, and reviewing third party applications. The servi tokens can not be transferred between the validators and its value becomes null once a validator creates a block.

The PoBelievability divides the validators in the network into two groups; a believable group and a normal group, based on the believability score. The normal group is then divided into sub groups randomly and each sub group is assigned to a validator from the believable group. The transactions are distributed randomly among the groups, where each believable validator checks the validity of the transactions and generates a block. The block is then appended on to the chain. This is done to increase the transaction throughput of the network by processing multiple transactions simultaneously. The believable validators are rewarded with mining fee along with the transactions fee. The block is then verified by the members of normal group assigned to that believable validator. If the believable validator is found to be malicious, it will lose all the crypto token it holds and the believability score will be zero. PoBelievability is currently implemented by the cryptocurrency platform internet of services token (IOST) [109] and has not been evaluated for security and privacy issues.

### 5.2.9. Proof of Authority (PoAuthority)

Proof of authority, a reputation-based consensus protocol was proposed in 2015 where the reputation of the miner is at stake instead of coins [110,111]. The role of a miner in PoAuthority is perfromed by a validator. The validators (known as authorities) in this algorithm are formally approved accounts whose identity is verified by an authorized public notary system and is kept public on-chain for cross-check. In order to be a validator, the authority has to have good reputation keeping them away from acting nefariously. Each validator will generate a block in a round-robin fashion. If a validator behaves malicious and proposes an invalid block, a negative reputation is attached to it. PoAuthority is used by cryptocurrency trading platforms POA network [112] and Vechain. There is no fee based incentive involved in PoAuthority, but the authorities are incentivized by attaching reputation to their identity. A variation of PoAuthority is Proof of Reputation (PoR) where instead of an authorized identity, a reputed organization is used as validator [113]. Once an organization passes the notary verification, it is designated as the authorized nodes in the network and the consensus proceeds in a way similar to PoAuthority. The reputation of an organization is measured using the market value of the organization, brand significance, and whether the organization is public or private. PoR is currently used by trading platform Gochain [113] and Menlo one [114]. PoAuthority and PoR algorithm makes the blockchain network less decentralized as the mining is performed by the fixed group of validators. Moreover, they have not been tested yet for its performance and protection against security threats.

### 5.2.10. Proof of Elapsed Time (PoET)

In order to solve the issues of rich getting richer and centralization of the network, Proof of Elapsed Time (PoET) was developed by Intel in 2016 [115] as a cost-efficient consensus protocol. PoET uses a trusted execution environment (TEE) [116] along with Intel's software guard extensions (SGX) [117], for fair and efficient leader election reducing the computation and energy cost and eliminating the wealth dominance. SGX ensures security by allowing applications to run a sensitive part of the code in a trusted environment without any modifications.

In PoET, each verifying network node sleeps after generating a random waiting time and the first node to complete the waiting time wins the right to generate the next block. The random waiting time is generated by running a code in the TEE using SGX which produces a signed attestation authenticating the execution of the code in a trusted environment. Each node in PoET generates a signed random wait time using the code executing in TEE and sleeps for that time period. The calculation of random wait time is done using the formula stated in equation 12 [118].

$$\text{WaitTime} = \text{MinimumWait} \; - \; \text{LocalAverageWait} \; \times \; log(r) \tag{12}$$

where MinimumWait is a fixed system parameter, LocalAverageWait is calculated using the number of active nodes in the network, and $r \in [0, 1]$ is a real number derived form the hash value of the node's previous signed attestation. The more the number of active nodes in the network, the more will be the waiting time of the nodes to avoid collision [118].

The first block to complete the wait time and wake up propagates a signed certificate to the network indicating that it has been selected as the leader. The remaining network nodes checks whether the nominated leader waited for the allocated waiting time, the allocated waiting time is not similar to the last 25 waiting times of that leader and the node is not consistently winning the election. A z-test is used to check consistent winning of a node. The test assumes that if the network has 'm' nodes, then each node has the same winning probability 'p', with the number of wins following a normal distribution N(mp, $\sqrt{mp(1-p)}$). The z-score for a node is calculated using equation 13 [118].

$$z = \frac{(\text{WinNum } - \text{mp})}{\sqrt{mp(1-p)}} \tag{13}$$

where WinNum is the number of blocks successfully created by that node. If the z-score is larger than a predefined value z-max, then the node will be not considered as the leader.

The leader creating a new block receives transactions fee. PoET has low energy consumption compared to PoW and does not favors the rich. However, PoET requires the use of specialized SGX hardware. Moreover, as the consensus depends on the SGX hardware developed by Intel, it makes Intel as the controlling authority making blockchain less decentralize. In addition, PoET is vulnerable to malicious attacks [119]. PoET is currently used by the HyperLedger Sawtooth blockchain platform [41].

#### 5.2.11. Proof of Activity (PoA)

In pure PoW the miners receive mining fee in addition to the transactions fee to encourage the miners to participate in the process of mining and thus securing the network. The current mining reward is 12.5 bitcoins and is halved after every 210,000 blocks mined. The PoW will become less significant when the mining reward will be obsolete and the miners will only rely on the transactions fee. Consequently, the miners dedicating their computing power may demand high transaction fees discouraging the use of the network. This is because, with low transaction fees the cost of mining will be much higher than the incentive received. This issue is known as the tragedy of commons [120] where everyone will selfishly look for their own benefit without contributing to the network security. Moreover, the validators validating the transactions in most of the consenus protocols receives no reward for their work. Proof of Authority (PoA) [121] developed in 2014, address this issues by using PoW in conjunction with PoS by dividing the transaction fees between a miner and the validators, encouraging more active participation of the nodes.

PoA in its first phase works as the pure PoW, where all the miners compete with each other to generate a block with a particular nonce in order to have the block's hash below a threshold and having predefined number of consecutive leading zeros. However, the block generated in PoA will not contain transactions. The miner broadcasts the created block to the network. In the second phase, PoA selects N validators referred as stakeholders based on the number of coins they have similar to that in PoS. Each selected stakeholder will verify and sign the block, and then broadcasts signed block in the network. The block will be signed by all the N-1 selected stakeholders until it reaches the $N^{th}$ stakeholder which includes the transactions in the block. The $N^{th}$ stakeholder will the hash the block and broadcasts it to the network. The transactions fee for the transaction included by the $N^{th}$ stakeholder is shared between the miner who created the block and the N stakeholders.

PoA fairly incentivizing all the block contributors in the network, suffers from the issue of high energy consumption as in PoW and it favors the rich as in PoS. PoA is not been tested much for its performance and security threats and is used by the cryptocurrency Decred [122].

In summary, capability based consensus protocols reduces the high energy consumption of the compute-intensive based protocols, but suffers from the issues of rich getting richer, encouragement of malicious activities, and network centralization. In order to address these issues, voting based consensus protocols were proposed in the literature.

### *5.3. Voting based Consensus Protocols*

The voting based consensus protocols uses a voting system to elect a miner for generating a block, eliminating the issue of high energy consumption of compute-intensive based protocols due to miner selection based on competitive approach and the issue of rich getting richer in capability based protocols due to the miner selection based on the wealth dominance. These protocols are designed to tolerate byzantine faults by assuming that there are independent node failures in the network or some of the nodes will behave maliciously. In reference to distributed systems, byzantine fault tolerance is the ability of the network to reach on a desired consensus despite of some of the nodes in the system are failing or behaving maliciously [123]. Voting based protocols are further classified into byzantine fault tolerance (BFT) based and crash fault tolerance (CFT) based. BFT-based consensus prevents the cases of failing node and malicious node. BFT is derived from the byzantine general's problem [124], a distributed computing network term for a situation where the network nodes must agree on a single state to avoid a complete failure, assuming that some of the nodes might be unreliable. The different BFT based protocols are practical byzantine fault tolerance, delegated byzantine fault tolerance, federated byzantine agreement, and combined delegated proof of stake and byzantine fault tolerance. On the other hand, CFT based consensus prevents only against the case of failing/crashing nodes. Different crash fault tolerance protocols are raft and federated.

### 5.3.1. Practical Byzantine Fault Tolerance (PBFT)

Practical byzantine fault tolerance was proposed by Castro et al. in the year 1999 [125]. In PBFT protocol, a group of nodes is being selected by a central authority with one node being the leader and the others as the backup nodes. All nodes in the system communicates with each other with the goal of reaching to an agreement by assuming that all honest nodes will have a exact same copy of the ledger. For the PBFT protocol to function correctly, the number of malicious or crashed nodes must not be equal or greater than $\frac{n}{3}$ out of the total n nodes at a given time. Thus, more are the number of nodes in the network, more unlikely it is for more than one third nodes being malicious. Consequently, more secure is the network.

Each round of generating a block in PBFT is known as a view and can be broken down into four phase in the context of blockchain. 1) A client sends request to the leader node to perform a transaction. 2) The leader node collects the transaction request and group them in a block. The block is then broadcasted to the backup nodes. 3) Each backup node will verify the transactions in the block and will create a block of valid transactions. The node then computes the hash of the block and broadcasts it to the other nodes. 4) A node waits for f+1 or two-third nodes to reply with the same hash, where f represents the number of faulty nodes. If the node receives the same reply, the block is then added to the ledger of that node.

PBFT is used by development platforms such as Hyperledger Fabric, Hyperledger Iroha, Oracle, Hydrachain, and BigchainDB. In Fabric and Oracle, the leader is referred as the orderer and the backup nodes are referred as the peers. In BigchainDB, PBFT is referred as tendermint protocol [126] where the nodes having stake (similar to PoS) are selected as the backup nodes. In tendermint, a backup node is termed as a validator, and the leader is elected from the group of validators in a round robin fashion each time a block is proposed. The transaction throughput of PBFT is better compared to that of PoW [127]. However, PBFT needs an authority for the selection of a leader and the backup nodes, making it less decentralized. For a private network having a selected group of backup nodes, PBFT performs satisfactorily eliminating the issues of compute-intensive and capability based protocols. However, when used by a public network with open participation for being a backup node, the communication becomes an issue. The more number of message transfers in PBFT may lead to increase in energy consumption due to network and communication overhead. Moreover, PBFT is prone to sybil attacks where one entity can create multiple faulty identities without any certificate authorization and can control a substantial fraction of the network [128] in a public network.

### 5.3.2. Delegated Byzantine Fault Tolerance (DBFT)

In order to avoid the centralized selection of leader and backup nodes in PBFT, Delegated byzantine fault tolerance was proposed by NEO cryptocurrency platform [129] in 2014 where the nodes are selected based on a voting process. In DBFT, a leader node is referred as a speaker and the backup nodes are referred as the delegates. The selection of the delegates is done by a voting system, where the network participants holding cryptocurrency participates in the process of voting. The weight of a vote by a participant is proportional to the amount of currency held by that participant. A speaker is elected randomly from the delegates. The process of block generation and validation takes places similar to that in PBFT. However, as the nodes are selected based on voting, there is a possibility where each participant votes for itself to be a delegate. In such situation where all the participants are selected as delegates, the network suffers from the issue of communication. For DBFT to function correctly, the total number of malicious or failing nodes in the network must not be equal to or greater than $\frac{(2n-1)}{3}$ out of the total n nodes in the network at a given time. The issues of high amount of communication data leading to increase in energy consumption and sybil attacks still prevails in DBFT.

### 5.3.3. Federated Byzantine Agreement (FBA)

In order to retain the decentralized property of blockchain and to avoid situations of communication overhead, Federated byzantine agreement was introduced in 2014 by Schwartz et al. [130] as a completely decentralized version of PBFT used by the Ripple network [131]. Comparing to PBFT, FBA does not require a list of selected nodes by a central authority to validate and process the transactions. FBA has a open membership like PoW, where any node in the network can participate in the process of consensus. A new transaction is added into the network if 80% of the nodes agree on the status of the transaction (which was 66% in PBFT). With all the nodes participating in the process of consensus and broadcasting the transaction status to each other in order to get 80% of the confirmation, the network will suffer from heavy communication. To address this issue each node in the network communicates with a list of nodes, known as unique node list (UNL) [18]. According to the ripple network, the intersection of UNL by any two nodes in different UNLs should be atleast one fifth of the total nodes in the network. This ensures that a transaction is validated by all the nodes in the network. Upon receiving transactions, each node will validate the transactions and will aggregate the valid transactions in a candidate list. Each node will boradcast its candidate list to all the nodes in the UNL. A transaction is considered valid if 80% of the nodes confirms that it is valid. The total number of malicious or failing nodes in the network should not exceed $\frac{(n-1)}{5}$ out of total n nodes in the network at a given time for FBA to function correctly [130]. FBA is also used by Stellar cryptocurrency platform where the UNL is referred as quorum slices [132]. In stellar, all the nodes in a quorum slice has to agree upon a transaction in order to update it to the ledger, unlike ripple where only 80% of the agreement was needed. Although, FBA makes the network decentralized and reduces the communication overhead, the protocol is more prone to malicious activities compared to DBFT and PBFT. This is because, the maximum number of faulty nodes required by FBA is $\frac{10n-5}{3n-3}$ and $\frac{5n}{3n-3}$ times less compared to DBFT and PBFT respectively.

### 5.3.4. Combined Delegated Proof of Stake and Byzantine Fault Tolerance (DPoS+BFT)

DPoS+BFT developed by the Credits blockchain platform [55] in 2018, uses DPoS consensus algorithm for the selection of the nodes participating in consensus process, and the BFT algorithm to update the ledger while protecting against malicious attacks. The working of DPoS+PBFT can be divided into two phases: 1) selection of nodes and 2) ledger update.

In the first phase, the algorithm selects the head nodes and the trusted nodes to participate in the process of consensus. The head nodes provides the confirmation of the transactions and creates a block, while the trusted nodes creates a list of valid transactions to add in the block using BFT. At each round of block creation, all the nodes in the network are allowed to participate for the selection as a head node or a trusted node. In order to get selected, the node should have an updated copy of the ledger. Each node in the network sends the hash of the last block in the ledger to the node who created the last block (previous head node). A predefined period of time is spared for this process. After this time period has elapsed the nodes which were unable to send the hash value are eliminated form the participation. The previous head node then removes the nodes having the hash value different from that of the last block, and prepares a list of eligible nodes. Each node in the list is assigned a random number and the list is arranged chronologically. The first node in the list is elected as the head node and a certain number of remaining nodes from the lit are selected as the trusted nodes. The number of trusted nodes is calculated using equation 14.

$$\text{Number of trusted nodes} = \begin{cases} 50\%, & \text{if m} \leq 200 \\ 100, & \text{if m} > 200 \end{cases} \tag{14}$$

where m in the total number of eligible nodes in the list. The list of head node and the trusted nodes in then broadcasted to all the nodes in the network.

In the second phase, the transactions proposed during the time the head and the trusted nodes were selected will then be sent to the new selected head node by all the nodes. The head node then generates a list of transactions and send it to all the trusted nodes. The trusted node will then validate each transaction and will create a list of valid transactions. Each trusted node will send its list of valid transactions to all other trusted nodes. Each trusted node will then create a list of approved transactions by selecting the transactions having majority of validation (using BFT) in all the lists received from each trusted node. Each node sends its list of approved transactions to other trusted nodes. A node with a mismatching list is eliminated considering it to be faulty or malicious. The list is then sent to the header node which creates the block of transactions in the list. The node then broadcasts this block to other nodes in the network for verification. A new round again begins with the first phase. The DPoS+BFT protocol increases the communication overhead tremendously as compared to the PBFT as the amount of data transfer between the trusted nodes is more than twice in DPoS+BFT as compared to the amount of data transfer between the backup nodes in PBFT, with the number of trusted and backup nodes being the same. Moreover, the amount of data transfer between the head nodes and trusted nodes in DPoS+BFT is more compared to that between the leader and the backup nodes in PBFT. Consequently, DPoS+BFT consumes more energy compared to PBFT.

### 5.3.5. Raft

The issue of communication overhead prevailing in the BFT based consensus protocols is eliminated in the CFT based algorithms by only allowing communication between the leader and the backup nodes and eliminating the communication among the backup nodes. Raft consensus algorithm proposed in 2014 by Ongaro et al. is a CFT based consensus ensuring safety only against the situation where a network node crashes without providing safety against malicious attacks [133]. The algorithm works correctly as far as more than 50% of the nodes are working normally at a given time in the network. At any given time each node in the network is in one of the following state: leader, follower or candidate. In the leader state, the node is responsible to generate the log entries of the transactions received by the clients. There is only one leader in the network. In the follower state, the node behaves passively and simply responds to the requests from the leader and the candidates. The requests in the network takes place using RPCs. AppendEntries RPCs are initiated by the leader for the replication of log and RequestVote PRCs are initiated by the candidates during elections. In the candidate state, the node elects a new leader. The working of raft consensus can be divided into three phases: 1) leader election, 2) log replication, and 3) Transaction execution.

- Leader election: Raft divides time into chunks of arbitary length known as terms. Each term is consecutively indexed and the election for the leader takes place with the beginning of a term. Initially a node is in a follower state and continues to receive AppendEntries RPCs from a previous leader. If the follower stops receiving the RPC request from the leader over a period of time known as election timeout, the follower assumes that there is no leader in the network for that term and begins a new election. To begin a new election, the follower increments the index of the current term and then transits into the candidate state. The follower now behaves as a candidate and participates in election. The node as a candidate then votes for itself to be the new leader and simultaneously issues RequestVote RPCs in parallel to all the other nodes. All the other candidate then respond to the request by submitting their vote. This is continued until one of the possible outcomes occur: (i) the candidate wins the election, (ii) another candidate wins the election, or (iii) no candidate is elected in that term.

(i) If the candidate receive votes from majority of the nodes in the network, then the candidate wins the election and sends a message to all other nodes stating its authority as the leader.

(ii) While waiting for the votes, a candidate may receive an authority message form another node claiming to be a leader. The candidate then compares the index value of the current term with that of the node sending the authority message. If the term index of the node is less than that of the candidate, the candidate will simply reject the message and if the index is more than that of the candidate, the candidate will recognize the node as a leader and will transit back to the follower state.

(iii) If many followers begin the election process simultaneously, the votes could split such that no candidate receives a majority of votes. In such situation each candidate will time out and start a new election process by incrementing the term index. In order to avoid split votes situation in the next election round, each candidate is allotted a random time out and the candidate with the shortest time out begins the election process.

- Log replication: Once a leader is elected it starts processing the transactions submitted by the clients. The leader validates each transaction and assigns a transaction index to all the valid transactions to maintain the order of the transaction. The leader the creates a block of this transactions and then issues AppendEntries request in parallel to all the follower nodes. The block is then replicated by each of the follower node followed by an acknowledgement to the leader regarding the replication.
- Transaction execution: When the leader receives majority of the acknowledgement response for the followers, it executes all the transactions in the block and then notifies the clients regarding the execution. This is known as block commitment. This is continued till the term ends.

Raft consensus algorithm solves the issue of crash tolerance but does not provide data integrity in the case when a node behaves maliciously. If the leader node behaves maliciously and executes invalid transactions then the entire blockchain becomes invalid. If the majority follower node behaves maliciously and does not send the replication acknowledgement message then the block can not be committed by the leader and the network suffers from the issue of latency. Raft consensus algorithm is used by the blockchain development platform Quorum [63].

#### 5.3.6. Federated

Federated CFT based consensus is proposed by the blockchain development platform chain core [134] where a leader and backup nodes are elected from a group of authorized nodes. The leader known as the block generator is responsible for validation of the transactions and creation of a block, and the backup nodes known as the block signers are responsible for verifying the blocks. The block generator is selected in a round-robin fashion at each block generation round. The transaction requests by the clients are sent to the block generators, which in turn validates all the transactions. The block generator then creates a block of valid transactions and broadcasts it to all the block signers in the network. Each block signer verifies the validity of the block and signs the block. The signed block is then sent back to the block generator. If the block generator receives M signatures ($\frac{N}{2}$ < M < N; N is the total number of block signers), it appends the block to its ledger and broadcasts the block to the network. All the network nodes will then verify that the block has enough signatures and update their own ledger. If the block does not contain required number of signatures, a new block generator is elected. The Federated protocol makes the network less decentralized as the selection of a leader and the backup nodes is done by a centralized authority.

In summary, the compute-intensive based consensus protocols suffers from the issues of high energy consumption, environmental pollution, low transaction throughput and less scalability, whereas the capability based protocols solves the issue of high energy consumption but tends to be biased towards the rich (wealth dominance) and more prone to malicious attacks. The voting based protocols solves the issues of high computational energy consumption, low transaction throughput, and scalability prevailing in the compute-intensive based protocols but they make the network less decentralized. Moreover, the number of data communication is high in voting based protocols which leads to energy consumption. Consequently, there exists a need for energy-efficient, decentralized, high transaction throughput, yet scalable blockchain consensus protocol to to address the misalignment between the existing protocols and the customer services where the applications are evolving rapidly in order to meet the requirements of the collaborative ecosystem. One possible solution is in the direction proposed by Tromp J. that uses cuckoo cycle-based PoW [135], an energy-efficient mining algorithm. The consensus uses the cuckoo hash table consisting of two same-sized tables each with its own hash function mapping a key to a table location, providing two possible locations for each key. The mining in cuckoo cycle then involves finding of a cuckoo graph cycle of a specific length, as denoted by the network difficulty. Moreover, Very few consensus protocols discussed in this section have been evaluated for its performance and security threats. However, these protocols are evaluated using different experimental environment and setup for different blockchain networks. For the moment there is no work in the literature comparing the performance evaluation of all the consensus algorithms in an unified experimental environment and setup.

## 6. Service Creation and Innovation Capabilities

Blockchain protocol initially designed for bitcoin, has shown its adaptability in various sectors beyond finance and banking without the need of a centralized trusted third party. The inherent characteristics of the technology such as local data access, fault tolerance, immutability, privacy, authenticity, and security has led government and large organizations for the adoption of the technology to improve the existing domains of applications like healthcare, finance, government, space, education, industries, media, and space. In this section, we provide a taxonomy of several existing blockchain applications and also discuss various potential future applications to introduce the vast horizon of the technology to the readers.

### *6.1. Existing Blockchain Applications*

Figure 9 shows a taxonomy of the existing blockchain applications classified based on their application domains. It also shows the type of blockchain network (public or private) used by these applications.

#### 6.1.1. Education

- Blockcerts: Blockcerts is a blockchain based application to create, issue, view and verify digital educational certificates [136,137]. This makes each individual capable of owning and sharing their own digital certificate on a peer-to-peer network. The network participants involve the university and the students.

#### 6.1.2. Medical and Healthcare

- MedRec: Management of Electronic Health Records (EHRs) has become complex with the patients moving across different medical organizations. With the mobility of the patients between different organizations and doctors, the medical data gets scattered and the accessed to medical history of a patient is lost. MedRec provides easy access and management of EHRs using the blockchain technology [138]. MedRec does not store the record directly; rather encodes the metadata containing the information about ownership, permission and the integrity of the data, that allows records to be accessed securely.
- SimplyVital Health: Care coordination of a patient receiving treatment among a team of care providers from therapist to specialist is difficult and complex. It is difficult to get analytics and

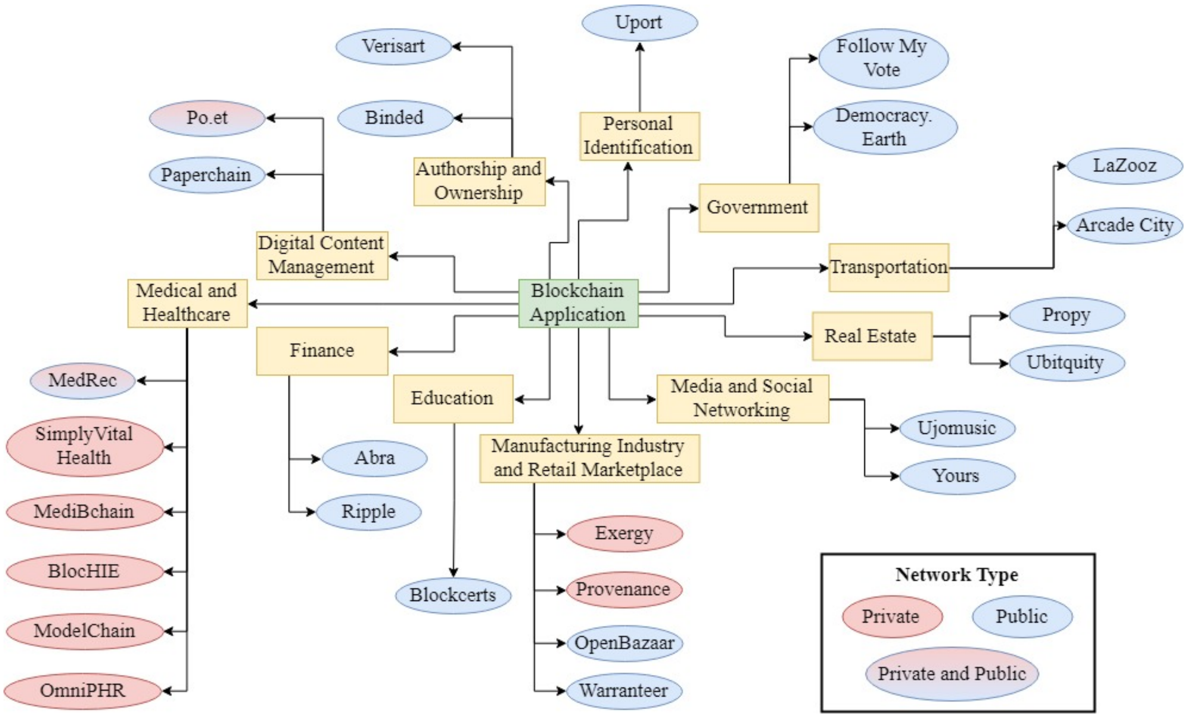

**Figure 9.** Existing Blockchain Applications in Different Domains.

insights into patient needs. SimplyVital Health different medical experts and organizations to share patients' medical data securely over a private blockchain network [139]. The copy of the ledger is maintained by the patients' care team. The platform uses machine learning algorithms to forecast best treatment cost for a patient in near real time to aid medical experts and organizations increase their earnings.

- MediBchain: A distributed healthcare data management system underpinned by blockchain that allows patients to store and share their personal health data [140]. The security of the data and the privacy of the patient is achieved by the blockchain characteristics. Moreover to tackle the issue of data breaching, the data is encrypted before storing in the blockchain.
- ModelChain: Blcochain-based framework to preserve the privacy of medical health data using machine learning techniques. It involves scalable proof of information algorithm to determine the order of online learning process [141].
- OmniPHR: A blockchain-based healthcare data management architecture to manage the scattered personal health data of patients [142]. The architecture allows patients to store and manage their data using blockchain to have a unified view and the data can be accessed by access control rights.
- BlocHIE: A blockchain-based platform for health information exchange to store and manage electronic medical records (EMRs) and personal health data (PHD) [143]. EMRs includes the medical report and diagnostic results prepared for a patient by a health professional whereas, PHD includes the patient data generated by sensors and health monitors. The framework uses two different blockchain to store EMRs and PHD. In order to reduce the network overhead, the medical data is stored locally by the doctor and the patient and the hash of the data is stored in blockchain to ensure privacy and authenticity of the data.

### 6.1.3. Finance

- Abra: A mobile application that allows users to to buy, sell, invest, and trade in different crypto and fiat currencies [144]. The cryptocurrency wallet of abra resides on the bitcoin blockchain network, with the miners and the validators maintaining the copy of the ledger. At the time of writing this paper, abra supports 30 different cryptocurrencies and 50 fiat currencies. The fiat currencies are used to buy or trade with cryptocurrencies or other fiat currencies via bank transfer/wire or credit/debit card.

- Ripple: A financial transaction application for currency exchange and remittance, powered by the blockchain technology featuring seamless global payment, and real-time transaction tracking, certainity and cost effectiveness. Ripple connects different banks and payment providers over the blockchain network for sending and receiving transactions [131]. Ripple also allows user to trade assets with the users in the network using the ripple native currency.

### 6.1.4. Manufacturing Industry and Retail Marketplace

- Exergy: A private blockchain platform that creates localized energy marketplaces for transacting energy across existing grid infrastructure [145]. Exergy allows prosumers generating energy through their own renewable resource to transact energy in near real-time with the consumers on the network. The platform connects prosumers and consumers in a network where the energy is generated, stored, and transacted locally creating more efficient and sustainable community. The exergy users and third parties verify each transactions in the network and maintain the copy of the ledger.
- Provenance: Provenance is a blockchain platform to help business developing trust by enhancing immutable and secure supply chain [146]. It records every product on the blockchain, allowing the consumers to trace and verify the origin, attributes and impact of the product. The network involves the producers, manufacturers, registrars, certifiers and auditors, and consumers with all the participants but the consumers maintaining the copy of ledger.
- OpenBazaar: An online peer-to-peer e-commerce that allow users to create their online store and to sell their products using cryptocurrency [147]. It is a bitcoin powered marketplace where the users and buyers can trade directly without requirement of any third party, eliminating the third party fees. OpenBazaar does not make use of fiat currency, thus eliminating the need of banks or other financial organizations.
- Warranteer: An application that allow users to save the warranty of their products electronically on a blockchain network [148]. The user upload the warranty of a product using the bar-code and then get notified for warranty status and expiration. The customers can get direct service from the service provider in the case of product malfunction by interacting to them over the network. Warranteer stores the data in cloud over the bitcoin network.

### 6.1.5. Media and Social Networking

- Ujomusic: A blockchain network which allows music creators to have the complete rights of the music they create [149]. It allows music creator to directly trade their music content to the fans using cryptocurrency. Ujomusic runs over the Ethereum blockchain network and the application has an embedded wallet to pay for the music content in ethers.
- Yours: A blockchain-based social networking application to share valuable information to the community [150]. The application has an embedded bitcoin wallet to pay the content creators and curators for their contribution. The application runs on the bitcoin blockchain network.

### 6.1.6. Real Estate

- Ubitquity: A blockchain-based platform that allows for transparent recording and tracking of title and property on a secure ledger [151]. The platform allows ownership of data without any involvement of a third trust party. It involves property owners, property buyers, real estate industry, and government authorities in the network.
- Propy: A blockchain based platform for overseas property transactions. The users can seel/buy property online from any country using either fiat currency or cryptocurrency, without invlovling third party [152]. All the closing documents related to a property transaction are added to the Ethereum blockchain network.

#### 6.1.7. Transportation

- Arcade City: A blockchain application where drivers can make local associations for ride sharing and car hiring underpinned by the Ethereum blockchain platform eliminating the need of a centralized cab service [153]. This will increase the income of the cab drivers by eliminating the intermediary fees by a centralized cab service such as Uber.
- LaZooz: A Decentralized Transportation Platform utilizing vehicles' unused space to create a variety of smart transportation solutions [154]. LaZooz platform will synchronize empty seats with transportation needs in real-time, matching like-minded people to create a great ride-sharing experience.

#### 6.1.8. Government

- Follow My Vote: An open source voting system using blockchain technology that connects all the voters and authorities in a peer-to -peer network [155]. This leads to a transparent election process making it trackable and 100% accurate.
- Democracy Earth: An open source blockchain based platform to develop incorruptible decision-making smart contracts for democratic affairs from local state level to country level to global world level, eliminating political intermediaries [156].

#### 6.1.9. Personal Identification

- Uport: A blockchain application for identity management where individuals can take ownership of their identity, running on the Ethereum platform [157]. User can request and send credentials, secure keys and data and can also sign transactions.

#### 6.1.10. Authorship and Ownership

- Binded: A blockchain based copyright platform where photographers can protect their images without any sort of fees [158]. The photographers can do the copyright process by themselves without depending on a third party. Images get uploaded in the blockchain network and the network generates a copyright certificate to ensure the ownership in a transparent manner.
- Verisart: A blockchain application to generate tamper-proof digital certificates for artworks and collectibles [159]. Verisart helps the artist to protect artworks securing them by cryptography on a immutable blockchain ledger.

#### 6.1.11. Digital Content Management

- Paperchain: A blockchain-based decentralized marketplace for organizations, content creators, and media companies to monetize their content and to make it globally available [160].
- Po.et: A blockchain-based application to record time-stamped immutable information about content creation [161]. The publishers and content creator just need to register to prevent the issues of content theft and licensing.

### *6.2. Emerging Blockchain Applications*

There are many other application in different domains that are emerging based on the blockchain technology, which can be summarized as follows.

#### 6.2.1. Space

Blockchain technology because of its secure transparent nature and absence of any third party can be used for improved data management and space communications in various interplanetary space missions. Recently, National Aeronautics and Space Administration (NASA) has awarded a grant to the University of Akron, Ohio, to develop a blockchain based autonomous spacecraft system called resilient networking and computing paradigm (RNCP) as a first step towards blockchain adoption

[162]. The Ethereum based system will make use of the underlying smart contract technology to build an intelligent spacecraft to autonomously tackle the floating debris in the space in real time.

Space Decentral [163], a decentralized autonomous organization has announced a decentralized space program known as coral [164] to facilitate 3D printing on the lunar surface. Spacechain [165], a platform combining space and blockchain has recently announced to carry Qtum's [166] blockchain software technology to outer space using a CubeSat to develop a decentralized data distribution network.

#### 6.2.2. Finance

American Internal Group Inc. and IBM recently announced the completion of pilot program for the standard chartered bank to create smart contract based insurance policy for faster cross-border policy creation and execution [167]. The real time system will allow different entities such as companies, their units, and the insurers to simultaneously share all data and documents related to policy in a transparent peer-to-peer network.

#### 6.2.3. Education

Blockchain technology has various applications in the sector of education and learning at individual, institutional, group, national and international levels. Blockchain can be used to securely store digital records of the students [168,169], infrastructure security, transportation management for staff and students, smart contracts for the staff payment and students credits [170,171], human resource [172], data storing and sharing in libraries [173], research articles submission, verification and reviewing [174,175], inter-organizational data collection and analysis [176], digital accreditation of academic learning and achievements [177], and university fees payment.

#### 6.2.4. Internet of Things (IoT)

With the advent of IoT technology, the world is now connected through a network of sensors and devices communicating and exchanging data information. Blockchain technology complements IoT by providing data security, reliability and efficient management. Implementation of the public ledger will overcome the low storage capabilities of the IoT devices. Various works to develop secure light-weight IoT architecture based on the blockchain technology have been proposed in the literature [178–180]. Furthermore, blockchain can be used for smart appliances, smart contracts [181] and economic exchange between IoT devices and sensors.

#### 6.2.5. Governance

The accountability, safety, and automation provided by the blockchain technology to handle public records could eventually obstruct the issues of fraudulent and corruption, while making the government services more transparent and efficient. Blockchain technology aims at providing government services by developing applications focusing on the areas such as digital personal identity of the citizens [182], birth, death and marriage certificates registration, smart city development [183–185], virtual notary and proof of ownership [24], e-residency [186], land management [187], decentralized voting systems [188–190], record tracking for the refugees, supply chain for the weapons of mass destruction, and disaster management [191].

## 7. Blockchain Issues and Possible Solutions

### *7.1. Scalability*

The major issue that limits the growth of public blockchain is the scalability. With no rules and validation process required for a user to join a public blockchain network, the network is growing rapidly in terms of users and data. The increasing number of users refers to increasing number of validations required for a transaction to be processed in a public blockchain. Consequently, decreasing the throughput of the network. Similarly, the increasing transaction data congest the network increasing the processing time as well as requires more data storage.

Many works in the literature are proposed to address the issue of scalability in the blockchain network. Segregated Witness (SegWit) [192], first introduced by Pieter Wuille aims to solve the issue of scalability by separating the transaction signature from the transaction data in the bitcoin network. Segwit separates the transaction signatures and the private key (generally the witness for a transaction) from the transaction data in a block and moves it to a side chain from the main chain. This increase the number of transactions in a block leading to an increase in throughput. Another solution to tackle the problem of scalability is lighting network [193]. Lightning network is a layer on top of the blockchain network that allows users to perform transaction off-chain using channels without waiting for the block to get processed. But, both of these solutions are only aimed to the bitcoin blockchain network.

To improve the scalability of the blockchain beyond bitcoin network, researchers at National University of Singapore developed zilliqa [194], a blockchain network based on the concept of sharding [195]. Zilliqa divides the nodes in the network into shards, and the the transaction are divided among the shards in parallel for verification. This increases the transaction throughput by the number of shard times as compared to the traditional blockchain network. However, all the possible solutions are still under development and are not tested for different existing blockchain networks.

### *7.2. High Energy Consumption*

The energy consumed by the compute-intensive based consensus protocols in the blockchain network is major concern for the environment. Most of the blockchain network uses the proof of work consensus algorithm as it is the most tested protocol ensuring high level of security. But, the protocol consumes a lot of energy while running the computing hardware in the process of mining. For example, considering the bitcoin network, a report by American magazine Grist stated that with the current mining power trend, the bitcoin network by 2020 will consumes the same amount of power that the entire world uses today. This energy consumption results in environmental hazard such as global warming and carbon foot prints [196].

Various researchers proposed the use of an alternate consensus protocol or use of energy-efficient hardware to tackle this issue. The compute-intensive based protocols are replaced by the capability or voting based protocols. Another solution is to use energy-efficient hardware that further complements the low energy consuming consensus protocol. The United States technology firm, Intel, proposed and patented a new generation hardware accelerator that aims to reduce the mining power of the bitcoin network by 35% [197].

Several efforts are being made to reduce the energy consumption of the blockchain as a technology while preserving its characteristics such as decentralization, privacy, and security. However, it would be interesting to study the use of sustainable energy for mining.

### *7.3. Throughput*

Every transaction in a blockchain network requires peer-to-peer verification before it can be processed. This becomes time consuming with more number of users, specially in a public blockchain network, where every user validates the transaction. Consequently, the number of transactions per second in a blockchain network pales the existing centralized systems for transaction.

Developers and researchers are working on a promising way to increase the performance of the blockchain technology even when it is scaled to global level. New consensus mechanisms such as byzantine fault tolerance, delegated proof of stake and federated byzantine agreement are being proposed and used by different blockchain platform development companies. Some researches proposed the use of linked transactions instead of linked blocks in the ledger. This is because the transactions grouped in a block are not processed until the block is mined by the miners, verified by the validators and updated in the ledger. Moreover, a block is not created unless a predefined block size is reached or a time period has elapsed. Consequently, the transactions suffers from the issue of latency. In an attempt to address this issue blockchain implementations are proposed in tier 3.0 where instead of creating a chain of creating blocks, the transactions are chained together. The transaction in the chain is linked to the hash of the previous transaction. The linked-transactions is implemented by the blockchain development platform Corda [198,199]. However, the ledger is shared only among the peers participating in the transactions resulting no peers on the network having identical copy of ledger. Consequently, when the ledger copy of one peer is corrupted or the peer behaves maliciously, it becomes difficult to agree on the correct state of the ledger. This leads to the issue of having a single point of failure in a centralized peer-to-peer network.

To solve the issue of single point of failure in linked-transactions ledger, blockchain development platform IOTA [200,201] proposed to use a ledger of transactions connected in a directed acyclic graph and maintained at each node. The directed graph of transactions are often referred to as tangle in IOTA, with the vertices of the graph representing the transactions and the edges representing the validation (also known as approval). The transaction which is not approved by any node is known as a tip. In order to get accepted in the network, a transaction has to select two tips for approval [202]. These tips are known as trunk transaction and branch transaction. However, the tangled transactions are more prone to sybil attack, because it is possible to create random transactions by an malicious attacker in order to select a tip, and can then process invalid transactions.

### 7.4. Cost and Complexity

The complexity of building and deploying a private blockchain network and the associated cost are major obstacles for the adoption of the technology. To address issue, cloud providers such as IBM [203], and Amazon [204] are providing cloud based blockchain templates to ease and automate the process of developing and deploying blockchain networks. Besides cloud based solution, there exists various offline blockchain development platforms (mentioned in this paper) which helps in easy blockchain development and deployment.

### 7.5. Data Privacy

The public blockchain network has the property of pseudo-anonymity, which makes all the transaction data over the network visible to public. While this feature helps in securing the network, but becomes a liability when to be used for a data sensitive application. Moreover, with multi-party transaction in the network, there are chances to track the real identity of a network participant [205].

Many technologies and methods such as use of deterministic wallets [206], ring signature mixtures [207], ring confidential transactions [208], channels [209], and private network have been proposed to achieve data privacy.

### 7.6. Lack of Governance

In a public blockchain network, where anyone can join the network, there rises the issue of lack of central governing authority to develop standard protocol and rules for transactions (specially financial transactions). The open source modification of the protocols makes it difficult to rely upon, when the majority of the network is dominated by malicious nodes. To solve this issue, organizations and companies are now moving on to private blockchain network, where a group of trusted members have the authority to modify the network.

### 7.7. Standardization and Interoperability

With the increasing attention towards the blockchain technology, various companies, developers, organizations and researchers are developing different blockchain platforms. Different platforms having different architecture, programming language, consensus protocols, and transaction flow makes it difficult for the interoperability of blockchain applications among various platforms.

Standardization could help organizations to develop platform independent applications and to share data between different blockchain networks. Seele is a blockchain platform enabling cross blockchain communications [210]. Openchain and Elements blockchain platforms provide this data transfer using a sidechain that interacts with the main blockchain. More work has to be done to develop specific standards for the development of platforms and consensus protocols to allow interoperability of the blockchain applications.

### *7.8. Access to External Data*

With increasing application of the blockchain technology linking it to the real world (IoT for example), it becomes necessary for the blockchain network to communicate with the data external to the network. Integrating service and relay network protocol [211] links the network nodes to the outside world [212]. However, there are still some issues to address such as:

- How to rely on the external data?
- There are chances where some of the nodes cannot reach the external data source due to issues in network connectivity or if the data source becomes unavailable. How to reach consensus in such situation?
- Who will govern the third party data sources?

## 8. Conclusions

The application of blockchain technology was introduced over a decade ago, in order to perform peer-to-peer transactions of digital currencies between a group of untrusted network participants without the need of a third party. Later, blockchain was evolved to develop multi-domain decentralized applications serving more than just financial transactions. Consequently, various blockchain architecture and consensus protocols proliferated. However, the need for an open, flexible, scalable, and energy-efficient has existed lately. This is due to the continuously evolving applications' needs in a collaborative ecosystem such as smart cities, social networking, governance, and smart heatlhcare with the ultimate goals of green computing and cost reduction. With the evolving application needs for better services, the current blockchain architecture and consensus protocols have misaligned with the goals for a green collaborative digital ecosystem.

Initially implemented using a public network allowing open participation and no data access control mechanism, blockchain suffered from the issues of scalability, energy-consumption, and security and privacy threats. Later, blockchain architectures were developed to enable development of blockchain applications using a public network addressing the issue of security and privacy. However, one of the major challenges in blockchain architecture and consensus protocols is the high amount of energy consumption to ensure security. The energy consumption of the compute-intensive based consensus protocols is escalating with the increasing number of participants having an adverse effect on the environment. One of the solution proposed was to use capability based or voting based protocols that reduce the energy hungry computation. However, attempts to tackle the issue of high energy consumption led to less scalable and decentralized architecture and consensus protocols. A good solution could be developing a compute-intensive based consensus protocol that is less computationally complex and energy efficient whenever a selection of a miner is involved.

On the other side, the inflexible and non-adaptive behavior of current architectures and consensus protocols act as barrier to serve the ultimate goals for a growing collaborative digital ecosystem. This is because current architectures and protocols target specific application domain without considering the future needs in an rapidly evolving collaborative ecosystem. Consequently, they fail behind flexible adaptation according to applications' needs and requiring the modifications in the underlying blockchain framework. In order to combat this, we need to develop modular and flexible architectures and consensus protocols that adapt to the dynamic nature of applications.

Furthermore, the security of the data and the privacy of the user's identity are issues of high relevance that can not be neglected while making the blockchain framework more energy-efficient and scalable. As of now, energy-efficient blockchain framework makes the network less decentralized and more prone to malicious attacks.

The objective of this article is to further the research about blockchain in context of smart cities collaborative ecosystem. In summary, it highlights the temporal evolution of the different blockchain architecture and consensus protocols, providing a retrospective analysis along with their contribution and limitations. Further research is still needed to develop a blockchain framework that is open to collaboration, reconfiguration, flexibility, scalability and energy-efficiency to address the gap between the existing architectures and protocols and the ultimate goals of green and cost-effective computing for better customer services.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Zheng, Z.; Xie, S.; Dai, H.N.; Wang, H. Blockchain challenges and opportunities: A survey. *Work Pap.–2016* **2016**.
2. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system **2008**.
3. Bitcoin mining consumes more electricity a year than Ireland | Technology | The Guardian. https://www.theguardian.com/technology/2017/nov/27/bitcoin-mining-consumes-electricity-ireland. (Accessed on 01/08/2019).
4. Stoll, C.; Klaaßen, L.; Gallersdörfer, U. The Carbon Footprint of Bitcoin. *Joule* **2019**. doi:https://doi.org/10.1016/j.joule.2019.05.012.
5. Ahamad, S.; Nair, M.; Varghese, B. A survey on crypto currencies. 4th International Conference on Advances in Computer Science, AETACS. Citeseer, 2013, pp. 42–48.
6. Tschorsch, F.; Scheuermann, B. Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies. *IEEE Communications Surveys Tutorials* **2016**, *18*, 2084–2123. doi:10.1109/COMST.2016.2535718.
7. Shen, C.; Pena-Mora, F. Blockchain for Cities—A Systematic Literature Review. *IEEE Access* **2018**, *6*, 76787–76819.
8. Al-Jaroodi, J.; Mohamed, N. Blockchain in Industries: A Survey. *IEEE Access* **2019**, *7*, 36500–36515. doi:10.1109/ACCESS.2019.2903554.
9. Jaoude, J.A.; Saade, R.G. Blockchain Applications–Usage in Different Domains. *IEEE Access* **2019**, *7*, 45360–45381.
10. Hölbl, M.; Kompara, M.; Kamišalić, A.; Nemec Zlatolas, L. A systematic review of the use of blockchain in healthcare. *Symmetry* **2018**, *10*, 470.
11. Conoscenti, M.; Vetro, A.; De Martin, J.C. Blockchain for the Internet of Things: A systematic literature review. 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA). IEEE, 2016, pp. 1–6.
12. Karafiloski, E.; Mishev, A. Blockchain solutions for big data challenges: A literature review. IEEE EUROCON 2017-17th International Conference on Smart Technologies. IEEE, 2017, pp. 763–768.
13. Yli-Huumo, J.; Ko, D.; Choi, S.; Park, S.; Smolander, K. Where is current research on blockchain technology?—a systematic review. *PloS one* **2016**, *11*, e0163477.
14. Li, X.; Jiang, P.; Chen, T.; Luo, X.; Wen, Q. A survey on the security of blockchain systems. *Future Generation Computer Systems* **2017**. doi:https://doi.org/10.1016/j.future.2017.08.020.
15. Feng, Q.; He, D.; Zeadally, S.; Khan, M.K.; Kumar, N. A survey on privacy protection in blockchain system. *Journal of Network and Computer Applications* **2019**, *126*, 45 – 58. doi:https://doi.org/10.1016/j.jnca.2018.10.020.
16. Park, J.; Park, J. Blockchain security in cloud computing: Use cases, challenges, and solutions. *Symmetry* **2017**, *9*, 164.
17. Mukhopadhyay, U.; Skjellum, A.; Hambolu, O.; Oakley, J.; Yu, L.; Brooks, R. A brief survey of Cryptocurrency systems. 2016 14th Annual Conference on Privacy, Security and Trust (PST), 2016, pp. 745–752. doi:10.1109/PST.2016.7906988.
18. Nguyen, T.; Kim, K. A survey about consensus algorithms used in Blockchain. *Journal of Information Processing Systems* **2018**, *14*, 101–128. doi:10.3745/JIPS.01.0024.

19. Wahab, A.; Mehmood, W. Survey of Consensus Protocols. *arXiv preprint arXiv:1810.03357* **2018**.
20. Remote procedure call - Wikipedia. https://en.wikipedia.org/wiki/Remote_procedure_call. (Accessed on 01/13/2019).
21. Web API - Wikipedia. https://en.wikipedia.org/wiki/Web_API. (Accessed on 01/13/2019).
22. Representational state transfer - Wikipedia. https://en.wikipedia.org/wiki/Representational_state_transfer. (Accessed on 01/13/2019).
23. Merkle, R.C. A Digital Signature Based on a Conventional Encryption Function. A Conference on the Theory and Applications of Cryptographic Techniques on Advances in Cryptology; Springer-Verlag: London, UK, UK, 1988; CRYPTO '87, pp. 369–378.
24. Swan, M. *Blockchain: Blueprint for a new economy*; " O'Reilly Media, Inc.", 2015.
25. Tian, F. An agri-food supply chain traceability system for China based on RFID & blockchain technology. Service Systems and Service Management (ICSSSM), 2016 13th International Conference on. IEEE, 2016, pp. 1–6.
26. Antonopoulos, A.M. *Mastering Bitcoin: unlocking digital cryptocurrencies*; " O'Reilly Media, Inc.", 2014.
27. Secure Hash Algorithms. https://en.wikipedia.org/wiki/Secure_Hash_Algorithms. (Accessed on 01/07/2019).
28. Brewer, E. CAP twelve years later: How the" rules" have changed. *Computer* **2012**, *45*, 23–29.
29. Girault, A.; Gössler, G.; Guerraoui, R.; Hamza, J.; Seredinschi, D.A. Why You Can't Beat Blockchains: Consistency and High Availability in Distributed Systems. *arXiv preprint arXiv:1710.09209* **2017**.
30. Decker, C.; Wattenhofer, R. Information propagation in the bitcoin network. Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on. IEEE, 2013, pp. 1–10.
31. Swan, M. *Blockchain: Blueprint for a New Economy*, 1st ed.; O'Reilly Media, Inc., 2015.
32. Szabo, N. Smart contracts: building blocks for digital markets. *EXTROPY: The Journal of Transhumanist Thought,(16)* **1996**.
33. Seele: Blockchain 4.0 or Marketing? - CoinAnnouncer. https://www.coinannouncer.com/seele-blockchain-4-0-or-marketing/. (Accessed on 01/03/2019).
34. Pseudonymity. https://en.wikipedia.org/wiki/Pseudonymity. (Accessed on 01/18/2019).
35. Joshi, A.P.; Han, M.; Wang, Y. A survey on security and privacy issues of blockchain technology. *Mathematical Foundations of Computing* **2018**, *1*, 121–147.
36. Buterin, V.; others. A next-generation smart contract and decentralized application platform. *white paper* **2014**.
37. Lai, R.; Chuen, D.L.K. Blockchain–from public to private. In *Handbook of Blockchain, Digital Finance, and Inclusion, Volume 2*; Elsevier, 2018; pp. 145–177.
38. Greenspan, G. MultiChain private blockchain—White paper. *URl: http://www. multichain. com/download/MultiChain-White-Paper. pdf* **2015**.
39. Hyperledger Burrow. https://www.hyperledger.org/projects/hyperledger-burrow. (Accessed on 12/29/2018).
40. Chain core. https://chain.com/docs/1.2/core/get-started/introduction. (Accessed on 12/29/2018).
41. Introduction — Sawtooth. https://sawtooth.hyperledger.org/docs/core/releases/1.0/introduction.html. (Accessed on 12/29/2018).
42. GitHub - HydraChain/hydrachain: Permissioned Distributed Ledger based on Ethereum. https://github.com/HydraChain/hydrachain. (Accessed on 01/03/2019).
43. Hyperledger Iroha – Hyperledger. https://www.hyperledger.org/projects/iroha. (Accessed on 12/29/2018).
44. The Burst Dymaxion. https://www.burst-coin.org/wp-content/uploads/2017/07/The-Burst-Dymaxion-1.00.pdf. (Accessed on 01/16/2019).
45. NEM white paper. https://nem.io/wp-content/themes/nem/files/NEM_techRef.pdf. (Accessed on 01/16/2019).
46. McConaghy, T.; Marques, R.; Müller, A.; De Jonghe, D.; McConaghy, T.; McMullen, G.; Henderson, R.; Bellemare, S.; Granzotto, A. BigchainDB: a scalable blockchain database. *white paper, BigChainDB* **2016**.
47. Quorum Overview. https://github.com/jpmorganchase/quorum/wiki/Quorum-Overview. (Accessed on 12/30/2018).

48. Durumeric, Z.; Kasten, J.; Bailey, M.; Halderman, J.A. Analysis of the HTTPS certificate ecosystem. Proceedings of the 2013 conference on Internet measurement conference. ACM, 2013, pp. 291–304.
49. Potlapally, N.R.; Ravi, S.; Raghunathan, A.; Jha, N.K. Analyzing the energy consumption of security protocols. Proceedings of the 2003 international symposium on Low power electronics and design. ACM, 2003, pp. 30–35.
50. Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; De Caro, A.; Enyeart, D.; Ferris, C.; Laventman, G.; Manevich, Y.; Muralidharan, S.; Murthy, C.; Nguyen, B.; Sethi, M.; Singh, G.; Smith, K.; Sorniotti, A.; Stathakopoulou, C.; Vukolić, M.; Cocco, S.W.; Yellick, J. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. Proceedings of the Thirteenth EuroSys Conference; ACM: New York, NY, USA, 2018; EuroSys '18, pp. 30:1–30:15. doi:10.1145/3190508.3190538.
51. Hyperledger Fabric – Hyperledger. https://www.hyperledger.org/projects/fabric. (Accessed on 12/29/2018).
52. Channels. https://hyperledger-fabric.readthedocs.io/en/release-1.3/channels.html. (Accessed on 12/29/2018).
53. Thakkar, P.; Nathan, S.; Vishwanathan, B. Performance Benchmarking and Optimizing Hyperledger Fabric Blockchain Platform. *arXiv preprint arXiv:1805.11390* **2018**.
54. Blockchain Platform | Oracle Cloud. https://cloud.oracle.com/en_US/blockchain. (Accessed on 01/14/2019).
55. Credits white paper. https://credits.com/Content/Docs/TechnicalWhitePaperCREDITSEng.pdf. (Accessed on 12/29/2018).
56. LZ77. https://en.wikipedia.org/wiki/LZ77_and_LZ78. (Accessed on 12/29/2018).
57. Huffman coding. https://en.wikipedia.org/wiki/Huffman_coding. (Accessed on 12/29/2018).
58. Dorri, A.; Kanhere, S.S.; Jurdak, R. Towards an optimized blockchain for IoT. Proceedings of the second international conference on Internet-of-Things design and implementation. ACM, 2017, pp. 173–178.
59. Kousaridas, A.; Falangitis, S.; Magdalinos, P.; Alonistioti, N.; Dillinger, M. SYSTAS: Density-based algorithm for clusters discovery in wireless networks. 2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC). IEEE, 2015, pp. 2126–2131.
60. Elements | elementsproject.org. https://elementsproject.org/. (Accessed on 12/31/2018).
61. Overview of Openchain. https://docs.openchain.org/en/latest/general/overview.html. (Accessed on 12/29/2018).
62. Home | Lisk Documentation. https://lisk.io/documentation/home. (Accessed on 01/14/2019).
63. Quorum white paper. https://github.com/jpmorganchase/quorum-docs/blob/master/Quorum%20Whitepaper%20v0.1.pdf. (Accessed on 12/30/2018).
64. Dwork, C.; Naor, M. Pricing via Processing or Combatting Junk Mail. Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology; Springer-Verlag: London, UK, UK, 1993; CRYPTO '92, pp. 139–147.
65. Eastlake 3rd, D.; Jones, P.; US, S.H.A. 1 (SHA1). Technical report, RFC Editor, 2001.
66. Brute-force search - Wikipedia. https://en.wikipedia.org/wiki/Brute-force_search. (Accessed on 01/18/2019).
67. Hashcash - Wikipedia. https://en.wikipedia.org/wiki/Hashcash. (Accessed on 12/18/2018).
68. The Mystery Behind Block Time – FACILELOGIN. https://medium.facilelogin.com/the-mystery-behind-block-time-63351e35603a. (Accessed on 12/18/2018).
69. of Commerce, U.D.; of Standards, N.I.; Technology. *Secure Hash Standard - SHS: Federal Information Processing Standards Publication 180-4*; CreateSpace Independent Publishing Platform: USA, 2012.
70. Bitcoin - Wikipedia. https://en.wikipedia.org/wiki/Bitcoin. (Accessed on 01/08/2019).
71. Gervais, A.; Karame, G.O.; Wüst, K.; Glykantzis, V.; Ritzdorf, H.; Capkun, S. On the security and performance of proof of work blockchains. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2016, pp. 3–16.
72. Schrijvers, O.; Bonneau, J.; Boneh, D.; Roughgarden, T. Incentive compatibility of bitcoin mining pool reward functions. International Conference on Financial Cryptography and Data Security. Springer, 2016, pp. 477–498.
73. Kroll, J.A.; Davey, I.C.; Felten, E.W. The economics of Bitcoin mining, or Bitcoin in the presence of adversaries. Proceedings of WEIS, 2013, Vol. 2013, p. 11.

74. Conti, M.; Kumar, E.S.; Lal, C.; Ruj, S. A Survey on Security and Privacy Issues of Bitcoin. *IEEE Communications Surveys Tutorials* **2018**, *20*, 3416–3452. doi:10.1109/COMST.2018.2842460.
75. Bitcoin Energy Consumption Index - Digiconomist. https://digiconomist.net/bitcoin-energy-consumption. (Accessed on 01/08/2019).
76. Litecoin - Open source P2P digital currency. https://litecoin.org/. (Accessed on 12/18/2018).
77. Dogecoin. https://dogecoin.com/. (Accessed on 12/18/2018).
78. King, S. Primecoin: Cryptocurrency with prime number proof-of-work. *July 7th* **2013**.
79. Young, A.; Yung, M. Finding length-3 positive cunningham chains and their cryptographic significance. International Algorithmic Number Theory Symposium. Springer, 1998, pp. 289–298.
80. Ingham, A.E.; Ingham, A.E. *The distribution of prime numbers*; Number 30, Cambridge University Press, 1990.
81. Forbes, T. Prime clusters and Cunningham chains. *Mathematics of Computation of the American Mathematical Society* **1999**, *68*, 1739–1747.
82. Ribenboim, P. *The new book of prime number records*; Springer Science & Business Media, 2012.
83. Cladwell, C. Fermat primality test. https://primes.utm.edu/prove/prove2_1.html. (Accessed on 01/10/2019).
84. lifchitz, H. Generalization of Euler-Lagrange theorem and new primality tests. http://www.primenumbers.net/Henri/us/NouvTh1us.htm. (Accessed on 01/10/2019).
85. Primecoin. http://primecoin.io/. (Accessed on 01/10/2019).
86. Komodo white paper. https://komodoplatform.com/wp-content/uploads/2018/05/2018-05-09-Komodo-White-Paper-Full.pdf. (Accessed on 01/25/2019).
87. Confirmation - Bitcoin Wiki. https://en.bitcoin.it/wiki/Confirmation. (Accessed on 01/24/2019).
88. Proof of stake instead of proof of work. https://bitcointalk.org/index.php?topic=27787.0. (Accessed on 01/10/2019).
89. King, S.; Nadal, S. PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake **2018**.
90. Whitepaper:Nxt - Nxt Wiki. http://nxtwiki.org/wiki/Whitepaper:Nxt. (Accessed on 01/10/2019).
91. Larimer, D. Delegated Proof-of-Stake Consensus. https://bitshares.org/technology/delegated-proof-of-stake-consensus, 2014. (Accessed on 01/23/2019).
92. bitshares.foundation/BitSharesBlockchain.pdf at master · bitshares-foundation/bitshares.foundation · GitHub. https://github.com/bitshares-foundation/bitshares.foundation/blob/master/download/articles/BitSharesBlockchain.pdf. (Accessed on 01/23/2019).
93. Nano – an instant, zero-fee, scalable currency. https://nano.org/en. (Accessed on 01/23/2019).
94. Cardano - Home of the Ada cryptocurrency and technological platform. https://www.cardano.org/en/home/. (Accessed on 01/23/2019).
95. Ren, L. Proof of stake velocity: Building the social currency of the digital age. *Self-published white paper* **2014**.
96. Reddcoin Social Currency - Official website. https://reddcoin.com/. (Accessed on 01/10/2019).
97. Proof of burn - Bitcoin Wiki. https://en.bitcoin.it/wiki/Proof_of_burn. (Accessed on 01/10/2019).
98. slimcoin whitepaper. http://www.doc.ic.ac.uk/~ids/realdotdot/crypto_papers_etc_worth_reading/proof_of_burn/slimcoin_whitepaper.pdf. (Accessed on 01/10/2019).
99. Slimcoin | A cryptocurrency for the long term. http://slimco.in/. (Accessed on 01/10/2019).
100. TGcoin. https://trade.tgco.in/. (Accessed on 01/10/2019).
101. Dziembowski, S.; Faust, S.; Kolmogorov, V.; Pietrzak, K. Proofs of space. Annual Cryptology Conference. Springer, 2015, pp. 585–605.
102. Bresson, E.; Canteaut, A.; Chevallier-Mames, B.; Clavier, C.; Fuhr, T.; Gouget, A.; Icart, T.; Misarsky, J.F.; Naya-Plasencia, M.; Paillier, P.; others. Shabal, a submission to NIST's cryptographic hash algorithm competition. *Submission to NIST* **2008**.
103. Space Coin. http://spacecoin.info/. (Accessed on 01/10/2019).
104. Home - Chia Network. https://chia.net/. (Accessed on 01/10/2019).
105. Burstcoin – The Linux of Blockchain. https://www.burst-coin.org/. (Accessed on 01/10/2019).
106. Solana: A new architecture for a high performance blockchain. https://solana.com/solana-whitepaper.pdf. (Accessed on 01/11/2019).
107. Proof of importance. https://nem.io/xem/harvesting-and-poi/. (Accessed on 01/18/2019).

108. Proof of Believability. https://github.com/iost-official/Documents/blob/master/Technical_White_Paper/EN/Tech_white_paper_EN.md. (Accessed on 01/18/2019).
109. IOST - UNLEASHING THE POWER OF BLOCKCHAIN. https://iost.io/. (Accessed on 01/18/2019).
110. Proof-of-authority - Wikipedia. https://en.wikipedia.org/wiki/Proof-of-authority. (Accessed on 01/23/2019).
111. wiki/Proof-of-Authority-Chains.md at master · paritytech/wiki · GitHub. https://github.com/paritytech/wiki/blob/master/Proof-of-Authority-Chains.md. (Accessed on 01/23/2019).
112. POA Network: public Ethereum sidechain with Proof of Autonomy consensus by independent validators. https://poa.network/. (Accessed on 01/23/2019).
113. gochain-whitepaper. https://gochain.io/gochain-whitepaper-v2.1.2.pdf. (Accessed on 01/23/2019).
114. Menlo One - Tools that make blockchain work for business. https://www.menlo.one/. (Accessed on 01/23/2019).
115. The second coming of blockchain | Intel® Software. https://software.intel.com/en-us/blogs/2017/02/14/the-second-coming-of-blockchain. (Accessed on 12/18/2018).
116. Sabt, M.; Achemlal, M.; Bouabdallah, A. Trusted Execution Environment: What It is, and What It is Not. 2015 IEEE Trustcom/BigDataSE/ISPA, 2015, Vol. 1, pp. 57–64. doi:10.1109/Trustcom.2015.357.
117. Intel SGX Homepage | Intel® Software. https://software.intel.com/en-us/sgx. (Accessed on 12/18/2018).
118. Chen, L.; Xu, L.; Shah, N.; Gao, Z.; Lu, Y.; Shi, W. On Security Analysis of Proof-of-Elapsed-Time (PoET). SSS, 2017.
119. Chen, L.; Xu, L.; Shah, N.; Gao, Z.; Lu, Y.; Shi, W. On security analysis of proof-of-elapsed-time (poet). International Symposium on Stabilization, Safety, and Security of Distributed Systems. Springer, 2017, pp. 282–297.
120. Hardin, G. The tragedy of the commons. *science* **1968**, *162*, 1243–1248.
121. Bentov, I.; Lee, C.; Mizrahi, A.; Rosenfeld, M. proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract] y. *ACM SIGMETRICS Performance Evaluation Review* **2014**, *42*, 34–37.
122. Decred - Autonomous Digital Currency. https://www.decred.org/. (Accessed on 01/11/2019).
123. Driscoll, K.; Hall, B.; Sivencrona, H.; Zumsteg, P. Byzantine fault tolerance, from theory to reality. International Conference on Computer Safety, Reliability, and Security. Springer, 2003, pp. 235–248.
124. Lamport, L.; Shostak, R.; Pease, M. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)* **1982**, *4*, 382–401.
125. Castro, M.; Liskov, B.; others. Practical Byzantine fault tolerance. OSDI, 1999, Vol. 99, pp. 173–186.
126. Kwon, J. Tendermint: Consensus without mining. *Draft v. 0.6, fall* **2014**.
127. Vukolić, M. The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication. International Workshop on Open Problems in Network Security. Springer, 2015, pp. 112–125.
128. Douceur, J.R. The sybil attack. International workshop on peer-to-peer systems. Springer, 2002, pp. 251–260.
129. NEO White Paper. https://docs.neo.org/en-us/whitepaper.html. (Accessed on 01/19/2019).
130. Schwartz, D.; Youngs, N.; Britto, A.; others. The Ripple protocol consensus algorithm. *Ripple Labs Inc White Paper* **2014**, *5*.
131. Ripple - One Frictionless Experience To Send Money Globally | Ripple. https://ripple.com/. (Accessed on 12/29/2018).
132. Mazieres, D. The stellar consensus protocol: A federated model for internet-level consensus. *Stellar Development Foundation* **2015**.
133. Ongaro, D.; Ousterhout, J. In Search of an Understandable Consensus Algorithm. 2014 USENIX Annual Technical Conference (USENIX ATC 14). USENIX Association, 2014, pp. 305–319.
134. Federated Consensus. https://chain.com/docs/1.2/protocol/papers/federated-consensus. (Accessed on 01/23/2019).
135. Tromp, J. Cuckoo Cycle: a memory-hard proof-of-work system. *IACR Cryptology ePrint Archive* **2014**, *2014*, 59.
136. Blockcerts : The Open Standard for Blockchain Credentials. https://www.blockcerts.org/. (Accessed on 12/29/2018).

137. About - Blockcerts : The Open Standard for Blockchain Credentials. https://www.blockcerts.org/about.html. (Accessed on 12/29/2018).
138. MedRec. https://medrec.media.mit.edu/. (Accessed on 12/29/2018).
139. SimplyVital Health. https://www.simplyvitalhealth.com/. (Accessed on 12/29/2018).
140. Al Omar, A.; Rahman, M.S.; Basu, A.; Kiyomoto, S. Medibchain: A blockchain based privacy preserving platform for healthcare data. International conference on security, privacy and anonymity in computation, communication and storage. Springer, 2017, pp. 534–543.
141. Kuo, T.T.; Ohno-Machado, L. Modelchain: Decentralized privacy-preserving healthcare predictive modeling framework on private blockchain networks. *arXiv preprint arXiv:1802.01746* **2018**.
142. Roehrs, A.; da Costa, C.A.; da Rosa Righi, R. OmniPHR: A distributed architecture model to integrate personal health records. *Journal of biomedical informatics* **2017**, *71*, 70–81.
143. Jiang, S.; Cao, J.; Wu, H.; Yang, Y.; Ma, M.; He, J. Blochie: a blockchain-based platform for healthcare information exchange. 2018 IEEE International Conference on Smart Computing (SMARTCOMP). IEEE, 2018, pp. 49–56.
144. Mobile Bitcoin Wallet App - BTC & Cryptocurrency Wallet App | Abra. https://www.abra.com/. (Accessed on 12/29/2018).
145. The Future of Energy | Blockchain, Transactive Grids, Microgrids, Energy Trading. https://lo3energy.com/. (Accessed on 12/29/2018).
146. Every product has a story | Provenance. https://www.provenance.org/. (Accessed on 12/29/2018).
147. OpenBazaar. https://openbazaar.org/. (Accessed on 12/29/2018).
148. About Us - Warranteer. http://www.warranteer.com/about-us. (Accessed on 12/29/2018).
149. Ujo Music. https://ujomusic.com/. (Accessed on 12/29/2018).
150. Home | Yours. https://www.yours.org/. (Accessed on 12/29/2018).
151. UBITQUITY - The First Enterprise Ready Blockchain-Secured Platform for Real Estate Recordkeeping | One Block At A Time. https://www.ubitquity.io/. (Accessed on 12/29/2018).
152. Propy - Buy or sell investment properties. https://propy.com/. (Accessed on 12/29/2018).
153. Arcade City. https://arcade.city/. (Accessed on 12/29/2018).
154. LaZooz. http://lazooz.org/. (Accessed on 12/29/2018).
155. The Online Voting Platform of The Future - Follow My Vote. https://followmyvote.com/. (Accessed on 12/29/2018).
156. Democracy Earth Foundation. https://www.democracy.earth/. (Accessed on 12/29/2018).
157. uPort.me. https://www.uport.me/. (Accessed on 12/29/2018).
158. Binded: Copyright made simple. https://binded.com/. (Accessed on 12/29/2018).
159. Verisart. https://verisart.com/. (Accessed on 12/29/2018).
160. Get Paperchain — Digital media's first global marketplace for AR financing. https://www.paperchain.io/. (Accessed on 12/29/2018).
161. Po.et - The decentralized protocol for content ownership, discovery and monetization of media. https://www.po.et/. (Accessed on 12/29/2018).
162. NASA Fund Researches the Potential of Blockchain Technology in Space. https://news.coinsquare.com/digital-currency/nasa-research-blockchain-in-space/. (Accessed on 12/29/2018).
163. Space Decentral: A Decentralized Autonomous Space Agency. Technical report.
164. Coral. https://medium.com/spacedecentral/introducing-coral-an-open-lunar-space-program-702e293c9869. (Accessed on 12/29/2018).
165. SpaceChain - Community-based Space Platform. https://spacechain.com/. (Accessed on 12/29/2018).
166. Qtum. https://qtum.org/en/. (Accessed on 12/29/2018).
167. Smart insurance policy powered by Blockchain. https://www.ibm.com/think/fintech/aig-ibm-standard-chartered-deliver-first-multinational-insurance-policy-powered-by-blockchain/. (Accessed on 12/29/2018).
168. IBM News room - 2017-08-09 Sony and Sony Global Education Develop a New System to Manage Students' Learning Data, Built on IBM Blockchain - United States. https://www-03.ibm.com/press/us/en/pressrelease/52970.wss. (Accessed on 12/19/2018).

169. Sharples, M.; Domingue, J. The blockchain and kudos: A distributed system for educational record, reputation and reward. European Conference on Technology Enhanced Learning. Springer, 2016, pp. 490–496.

170. Imagining a Blockchain University | Tom Vander Ark. https://www.gettingsmart.com/2018/06/imagining-a-blockchain-university/. (Accessed on 12/19/2018).

171. Turkanović, M.; Hölbl, M.; Košič, K.; Heričko, M.; Kamišalić, A. EduCTX: A blockchain-based higher education credit platform. *IEEE Access* **2018**, *6*, 5112–5127.

172. ChronoBank.io. https://chronobank.io/. (Accessed on 12/19/2018).

173. Use for Blockchain in Libraries. https://ischoolblogs.sjsu.edu/blockchains/blockchains-applied/applications/. (Accessed on 12/19/2018).

174. Spearpoint, M. A proposed currency system for academic peer review payments using the Blockchain Technology. *Publications* **2017**, *5*, 19.

175. Gipp, B.; Breitinger, C.; Meuschke, N.; Beel, J. CryptSubmit: Introducing Securely Timestamped Manuscript Submission and Peer Review Feedback Using the Blockchain. 2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL), 2017, pp. 1–4. doi:10.1109/JCDL.2017.7991588.

176. Bore, N.; Karumba, S.; Mutahi, J.; Darnell, S.S.; Wayua, C.; Weldemariam, K. Towards Blockchain-enabled School Information Hub. Proceedings of the Ninth International Conference on Information and Communication Technologies and Development. ACM, 2017, p. 19.

177. Grech, A.; Camilleri, A.F. Blockchain in education, 2017.

178. Dorri, A.; Kanhere, S.S.; Jurdak, R.; Gauravaram, P. Blockchain for IoT security and privacy: The case study of a smart home. 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), 2017, pp. 618–623. doi:10.1109/PERCOMW.2017.7917634.

179. Dorri, A.; Kanhere, S.S.; Jurdak, R. Towards an Optimized BlockChain for IoT. 2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI), 2017, pp. 173–178.

180. Kravitz, D.W.; Cooper, J. Securing user identity and transactions symbiotically: IoT meets blockchain. Global Internet of Things Summit (GIoTS), 2017. IEEE, 2017, pp. 1–6.

181. Christidis, K.; Devetsikiotis, M. Blockchains and Smart Contracts for the Internet of Things. *IEEE Access* **2016**, *4*, 2292–2303. doi:10.1109/ACCESS.2016.2566339.

182. Rivera, R.; Robledo, J.G.; Larios, V.M.; Avalos, J.M. How digital identity on blockchain can contribute in a smart city environment. 2017 International Smart Cities Conference (ISC2), 2017, pp. 1–4. doi:10.1109/ISC2.2017.8090839.

183. Ibba, S.; Pinna, A.; Seu, M.; Pani, F.E. CitySense: blockchain-oriented smart cities. Proceedings of the XP2017 Scientific Workshops. ACM, 2017, p. 12.

184. Biswas, K.; Muthukkumarasamy, V. Securing Smart Cities Using Blockchain Technology. 2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2016, pp. 1392–1393. doi:10.1109/HPCC-SmartCity-DSS.2016.0198.

185. Sharma, P.K.; Moon, S.Y.; Park, J.H. Block-VN: A Distributed Blockchain Based Vehicular Network Architecture in Smart City. *JIPS* **2017**, *13*, 184–195.

186. Sullivan, C.; Burger, E. E-residency and blockchain. *Computer Law & Security Review* **2017**, *33*, 470 – 481. doi:https://doi.org/10.1016/j.clsr.2017.03.016.

187. Pichel, F. Blockchain for land administration. *GIM International* **2016**, *30*, 38–39.

188. Meter, C. Design of Distributed Voting Systems. *CoRR* **2017**, *abs/1702.02566*.

189. Noizat, P. Chapter 22 - Blockchain Electronic Vote. In *Handbook of Digital Currency*; Chuen, D.L.K., Ed.; Academic Press: San Diego, 2015; pp. 453 – 461. doi:https://doi.org/10.1016/B978-0-12-802117-0.00022-9.

190. Hjalmarsson, F.P.; Hreioarsson, G.K.; Hamdaqa, M.; Hjalmtysson, G. Blockchain-Based E-Voting System. 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), 2018, Vol. 00, pp. 983–986. doi:10.1109/CLOUD.2018.00151.

191. Panesir, M.S. Blockchain Applications for Disaster Management and National Security. PhD thesis, 2018.

192. SegWit. https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki. (Accessed on 12/31/2018).

193. Poon, J.; Dryja, T. The bitcoin lightning network: Scalable off-chain instant payments. Technical report, Technical Report (draft). https://lightning. network, 2015.

194. Zilliqa. https://docs.zilliqa.com/positionpaper.pdf. (Accessed on 12/31/2018).

195. Sharding. https://www.investopedia.com/terms/s/sharding.asp. (Accessed on 12/31/2018).
196. Bitcoin could cost us our clean-energy future | Grist. https://grist.org/article/bitcoin-could-cost-us-our-clean-energy-future/. (Accessed on 12/31/2018).
197. BITCOIN MINING HARDWARE ACCELERATOR WITH OPTIMIZED MESSAGE DIGEST AND MESSAGE SCHEDULER DATAPATH. http://appft.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&u=%2Fnetahtml%2FPTO%2Fsearch-adv.html&r=1&p=1&f=G&l=50&d=PG01&S1=20180089642.PGNR.&OS=dn/20180089642&RS=DN/20180089642. (Accessed on 12/31/2018).
198. Brown, R.G.; Carlyle, J.; Grigg, I.; Hearn, M. Corda: An introduction. *R3 CEV, August* **2016**.
199. Corda. https://docs.corda.net/. (Accessed on 12/30/2018).
200. Popov, S. The tangle. *cit. on* **2016**, p. 131.
201. What is IOTA. https://docs.iota.org/introduction/what-is-iota. (Accessed on 12/29/2018).
202. Tip selection - IOTA Docs. https://docs.iota.org/introduction/tangle/tip-selection. (Accessed on 12/29/2018).
203. Blockchain on Bluemix: IBM Blockchain Blog. https://www.ibm.com/blogs/blockchain/category/blockchain-on-cloud/. (Accessed on 01/25/2019).
204. Blockchain on AWS. https://aws.amazon.com/blockchain/. (Accessed on 01/25/2019).
205. Genkin, D.; Papadopoulos, D.; Papamanthou, C. Privacy in Decentralized Cryptocurrencies. *Commun. ACM* **2018**, *61*, 78–88. doi:10.1145/3132696.
206. Deterministic wallet - Bitcoin Wiki. https://en.bitcoin.it/wiki/Deterministic_wallet. (Accessed on 01/02/2019).
207. Ring signature - Wikipedia. https://en.wikipedia.org/wiki/Ring_signature. (Accessed on 01/02/2019).
208. Noether, S.; Mackenzie, A.; others. Ring confidential transactions. *Ledger* **2016**, *1*, 1–18.
209. Coleman, J.; Horne, L.; Xuanji, L. Counterfactual: Generalized state channels, 2018.
210. Seele. https://seele.pro/. (Accessed on 01/03/2019).
211. Hirn, M. Rlay: A Decentralized Information Network. https://rlay-project.github.io/rlay.com/rlay-whitepaper.pdf. (Accessed on 01/02/2019).
212. Corda nodes. https://docs.corda.net/corda-test-networks.html. (Accessed on 12/30/2018).