symmetry

MDPI

Article

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

Leila Ismail * and Huned Materwala

Department of Computer Science and Software Engineering, College of IT, United Arab Emirates University, 15551 Al Maqam, Al Ain, UAE; huned.m@uaeu.ac.ae

* Correspondence: leila@uaeu.ac.ae; Tel.: +971-3-7135530 (ext.5530)

Received: 25 July 2019; Accepted: 16 September 2019; Published: 24 September 2019

Abstract: Over the last decade, blockchain technology has emerged to provide solutions to the complexity and privacy challenges of using distributed databases. It reduces cost for customers by eliminating intermediaries and builds trust in peer-to-peer communications. 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-city ecosystem, healthcare, and governance. Many platforms, with different architectures and consensus protocols, have been introduced. Consequently, it becomes challenging for an application developer to choose the right platform. Furthermore, blockchain has misaligned with the goals for an 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 conducive to their development and contributions to building collaborative applications. We introduce a classification of those architectures helping developers to choose a suitable platform for applications and providing insights for future research directions in the field to build new frameworks.

Keywords: blockchain; consensus; hash functions; privacy; replication; scalability

1. Introduction

The blockchain is a disruptive technology that has emerged for decentralized applications as the outcome of complexity, security, and intermediaries extending across over half a century. Blockchain, a peer-to-peer system, enables users to maintain a ledger of transactions that is replicated and synchronized over multiple user servers [1]. The transactions are processed and verified by consensus of most of the network participants, eliminating the need for 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 for green and cost-efficient computing. Therefore, it becomes difficult for an application developer to choose the right platform. In addition, current blockchain architectures and consensus protocols have misaligned with the goals for a green collaborative decentralized and agile ecosystem. Consequently, it becomes increasingly vital to address this issue and build new frameworks to align blockchain with those goals. Our main aim in this paper is to help developers to select the right platform architecture and consensus protocol for their applications. Therefore, we classify the platforms architectures into categories based on the applications nature in terms of the number of ledgers and interoperability needs. We discuss the advantages of each category

and the inherent problems, and begin offering solutions towards a scalable, cost-efficient, and green blockchain framework.

Originally developed to transfer 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 difficult to foresee the uses of blockchain. In 2017, the Bitcoin mining used around 30.14 TeraWatt hours (TWh) of energy, which is equivalent to the energy usage of entire Ireland in a year [3]. 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 countries such as 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. A growing business requires trust and transparency between the customers and the business providers. Customers need to eliminate intermediaries to reduce the transaction cost. The issues of data communication overhead with increasing number of network participants further hinder the quality of services of the developed applications. Addressing these problems of energy consumption and scalability often trades off with security and privacy. Therefore, the goals of this paper are four-fold: (1) provide a temporal evolution of the blockchain 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 the application developers to choose the right platform architecture, (3) evaluate the current research on the topics of blockchain architecture and consensus protocols under comprehensive taxonomies, and address the challenges and the issues therein, and (4) use the taxonomies to guide future research 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 the 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 areas such as medical, finance, education, manufacturing industry and retail marketplace, media, real estate, transportation, government, authorship and ownership, and digital content management.
6. We provide critical analysis of the different issues prevailing in blockchain technology and the possible solutions that were proposed for these issues.
7. We propose directions toward the development of scalable and energy-efficient blockchain 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. The overview of blockchain and transaction execution flow along with an organizational framework are presented in Section 3. Sections 4 and 5 synthesize 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 in blockchain

along with possible solutions. We conclude the paper with possible future research directions in Section 8.

2. Related Surveys

While there has been a lot of attention paid towards the blockchain technology, there have been relatively few surveys conducted in this area. These surveys 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 the 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 the different blockchain use cases for smart cities and organize them into nine categories. The nine 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 the 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 of blockchain in healthcare. Conoscenti et al. [11] discussed the issues of scalability, integrity and privacy in blockchain for the 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 of blockchain technology and its applications. The authors concluded that there is a research gap in 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 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 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 the different threats such as double-spending, the 51% attack, the security for a private key, vulnerable and malicious activities, and data privacy in a blockchain network. Feng et al. [15] reviewed the privacy threats and provided a taxonomy of existing cryptographic defense mechanisms to address those threats. Park et al. [16] presented the security challenges and solutions in the Bitcoin network. The authors also proposed a solution to secure data in a cloud computing environment using blockchain.

For consensus protocols, Mukhopadhyay et al. [17] surveys the mining techniques in the 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. surveyed 7 consensus protocols [19] presenting their advantages and drawbacks. Nguyen et al. described 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. It is fine-grained which does not allow the addition of 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 rationales 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 categories computing power 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. We then present an organizational framework to help the readers to effectively design a blockchain architecture and to develop applications.

3.1. Blockchain Overview

Figure 1 shows an overview of blockchain technology. We divide the blockchain architecture into 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 types of nodes: simple (also referred to as a light node), full, and mining. 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 does. 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 transaction records. The platform layer facilitates Remote Procedure Calls (RPC) [20], web Application Programming Interface (API) [21], and REpresentational State Transfer (REST) API's [22] for the communication between the network participants.

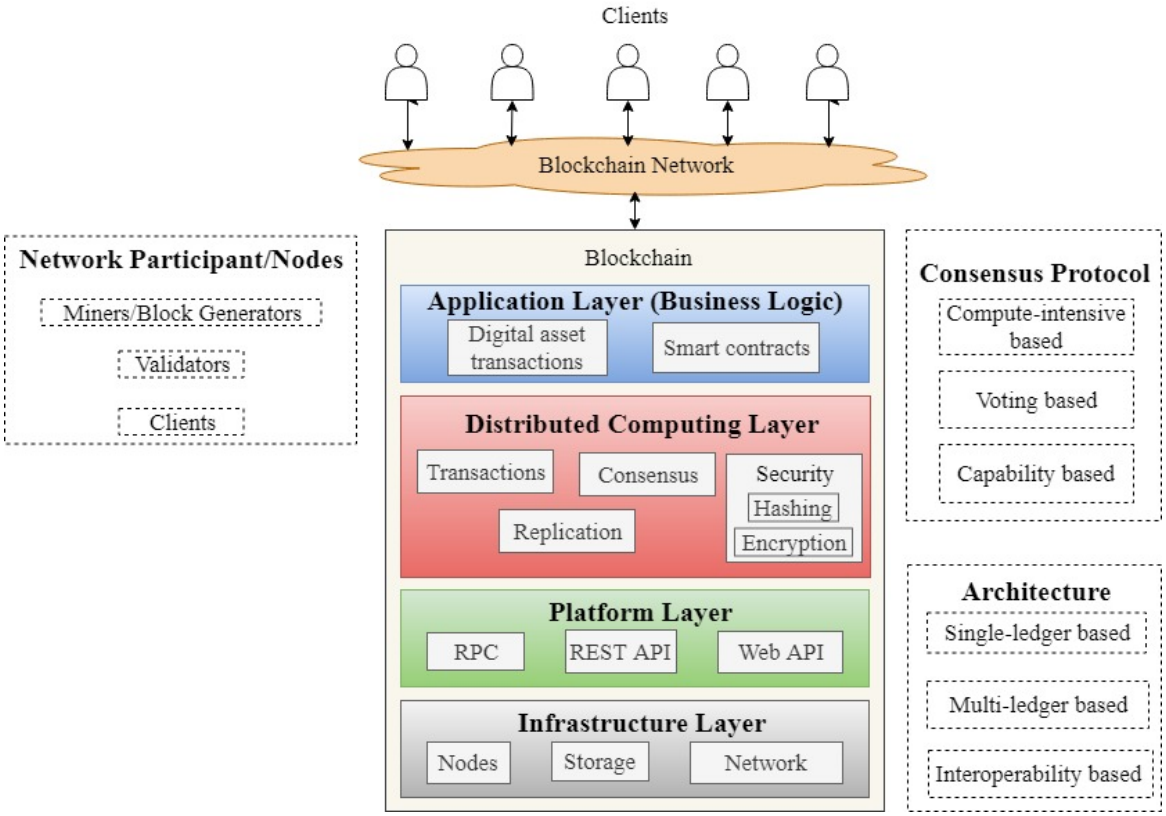

Figure 1. Overview of Blockchain.

The distributed computing layer ensures local access to data, fault tolerance, immutability, privacy, authenticity, and security for the transaction data. Immutability is the blockchain property that does not allow modification of the transaction records once updated 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 the next block generation. In addition, this layer is responsible for user authentication by using an encryption technique [23] and for data privacy via a hashing technique [24]. The application layer is the business logic for digital asset transactions

and the execution of smart contracts. An application developed on top of a blockchain network can be accessed by the clients using the platform layer. The layers of the blockchain architecture have the following characteristics.

- *Decentralization*: The transactions in blockchain are processed and validated by the consensus of most network nodes. They are replicated on the nodes in a ledger. This eliminates the need for an intermediary to share and maintain the transactions data [1].
- *Immutability*: The transactions in blockchain are stored into blocks. Each block in the chain is linked to the previous block using a cryptographic hash function. Any attempt to modify the content of a block will affect the subsequent blocks in the chain. Consequently, a malicious attacker needs to change all the succeeding blocks in the chain to modify a particular block, which is computationally difficult because the chained blocks are replicated over multiple nodes.
- *Transparency*: The ledger is only updated when most of the nodes reach a consensus. Changes in the network are publicly visible ensuring transparency and security.
- *Traceability*: The distributed and transparent nature of blockchain makes it easier to trace any transaction event. Each update in the state of an 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 block data. The header consists of the block metadata information such as the Merkle tree root hash, the previous block hash, the timestamp, 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 by 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 [25].
- *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 the previous block hash value in a block header is to ensure the immutability of the blockchain ledger.
- *Timestamp*: 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.

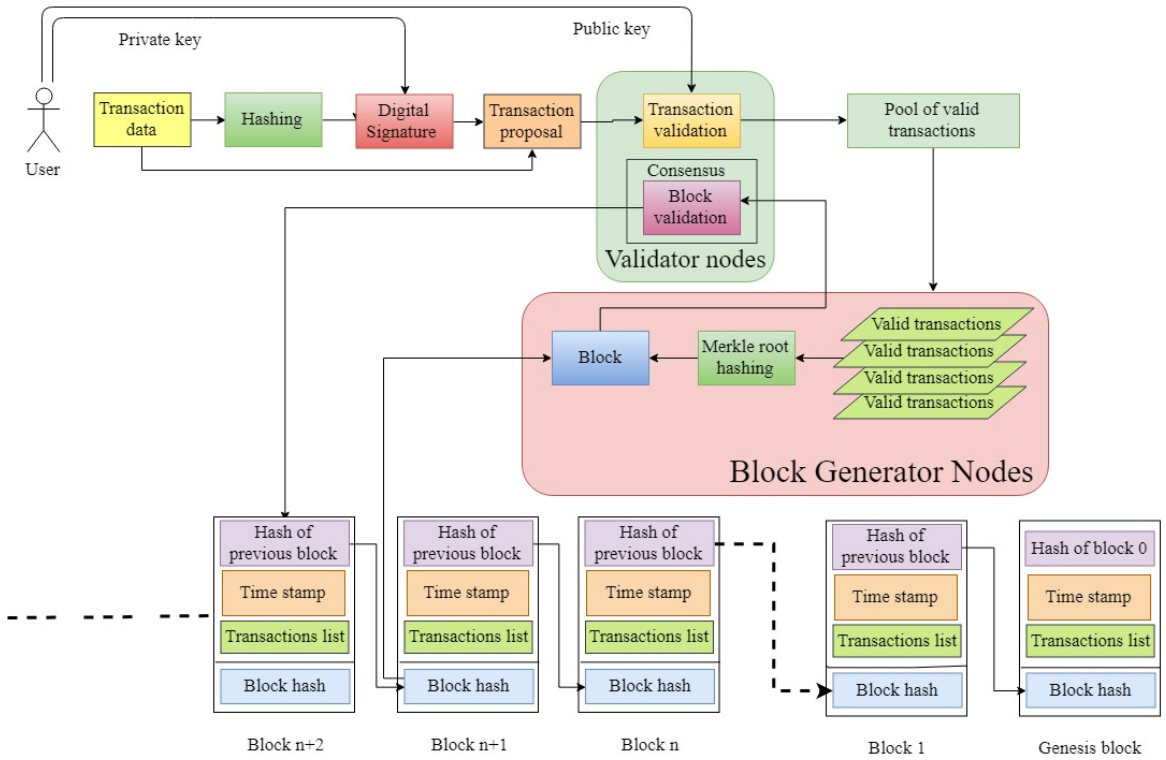

Figure 2. Overview of Transaction Execution Flow in Blockchain.

The execution flow consists of the following steps:

1. *Transaction proposal*: The user first hashes the transaction data using a hash function [26] 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 exceed 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 only requires the hash values of the Merkle path connecting the transaction to the Merkle root. Consequently, a node that does 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 if Merkle root is not used [25]. After calculating the Merkle root hash value, the block hash is generated. The miner broadcasts the block to the network. The validating nodes verify the validity of the block by checking the correctness of the followings: (1) the block hash, (2) the block timestamp is greater than the timestamp of the previous block, (3) the block height and size values, (4) the previous block hash value, and (5) the validity of all the transactions in the block. Each validating node appends the valid block to its own 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 high network latency. Ideally, the copy of the ledger should be consistent between the 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 [27] should be achieved. The replication strategy used by the blockchain network is the Monotonic Prefix Consistency (MPC) [28]. 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, forming a hierarchy of nodes.

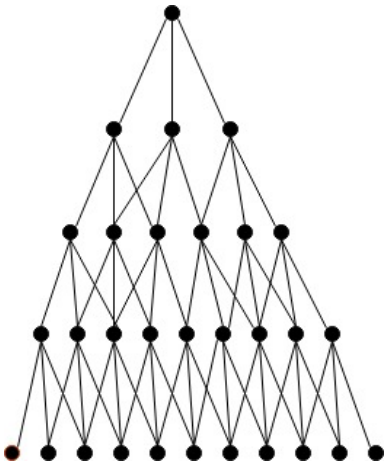

Figure 3. Hierarchy-based Connection of Nodes in the Blockchain Network.

The broadcast of transactions and blocks involves a high number of messages communicated over the network. To avoid communicating data to nodes which have already received it from some other node, the transaction and block data are not broadcasted directly to the nodes [29]. Instead, the node receiving a transaction or a block sends first 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 receives an invitation message and does not have the transaction or the block replies with a get data message [29].

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 customer trust in the network which helps a growing business for enterprises and individuals. Consequently, over the last decade there has been a proliferation of platforms and consensus protocols to develop applications in different domains. This makes it difficult for an application developer to choose 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 our classification 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.

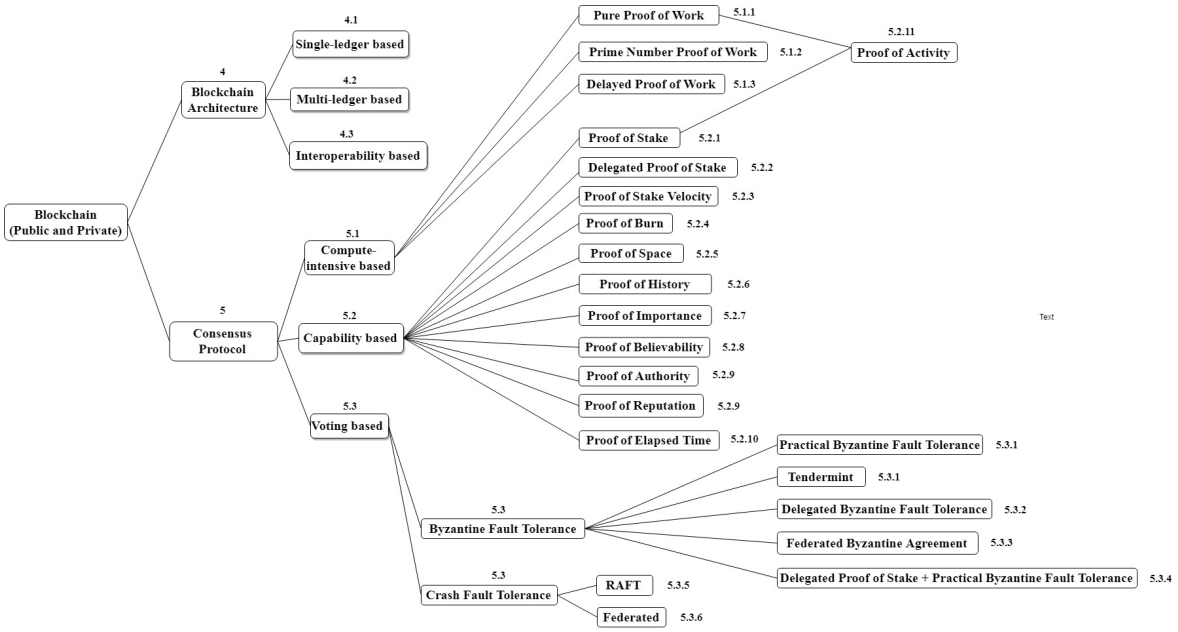

Figure 4. A Taxonomy-based Overview of Blockchain Literature Surveyed in this Paper. The numbers indicate the section/subsection in this paper.

The technology evolved over time due to its adoption by different types 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 [24]. Later in 2013, Tier 2.0 was introduced to facilitate the digital transfer of non-financial assets by using smart contracts [30]. 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 visible transactions data. Consequently, Tier 3.0 was introduced in 2015 for the development of applications in a private network. Due to the need for interoperability in a rapidly evolving collaborative ecosystem, Tier 4.0 was initiated in 2018 [31].

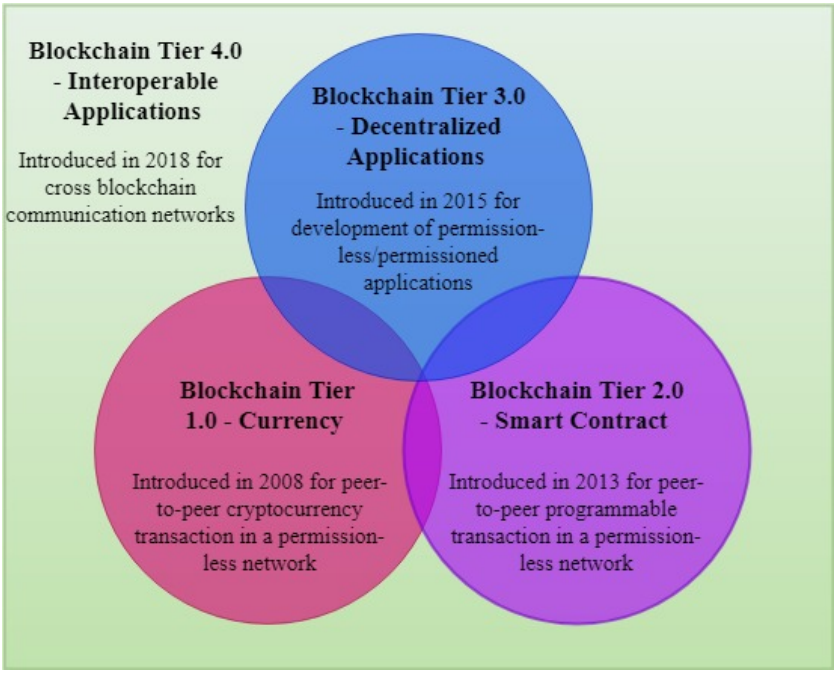

Figure 5. Blockchain Tiers.

Blockchain architecture is either public-centric or private-centric. A public blockchain network, also called a permission-less blockchain, allows anyone to join the network without permission [1]. The user can join as a simple node, a validating node or a mining/block generating node. This type of network typically offers an incentive for the users to participate in the consensus to encourage more participants to join. The identity of a network participant is pseudo-anonymous via a pseudo-name [32] by using a public key. The transaction data is public leading to the issue of data privacy [33]. Private blockchain, known also permissioned, is an invite-only network by an authentication authority [1]. The network involves access-control rights for ledger queries and updates. Table 1 shows the comparison between the public and the private blockchains.

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

4. Evolution and Taxonomy of Blockchain Architectures

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

4.1. Single-Ledger-Based Architecture

Single-ledger-based platforms were developed in Tier 1.0 and then remained in Tiers 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 [34]. As shown in Figure 6, the network participants are represented by peers (or nodes). A node can be simple, full, or mining. A client, the user issuing the transactions, uses the RPC to connect to the blockchain and an integration service to connect to an external system. An external system 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. If the external system is malicious, the validity of the transaction becomes questionable [35]. 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.

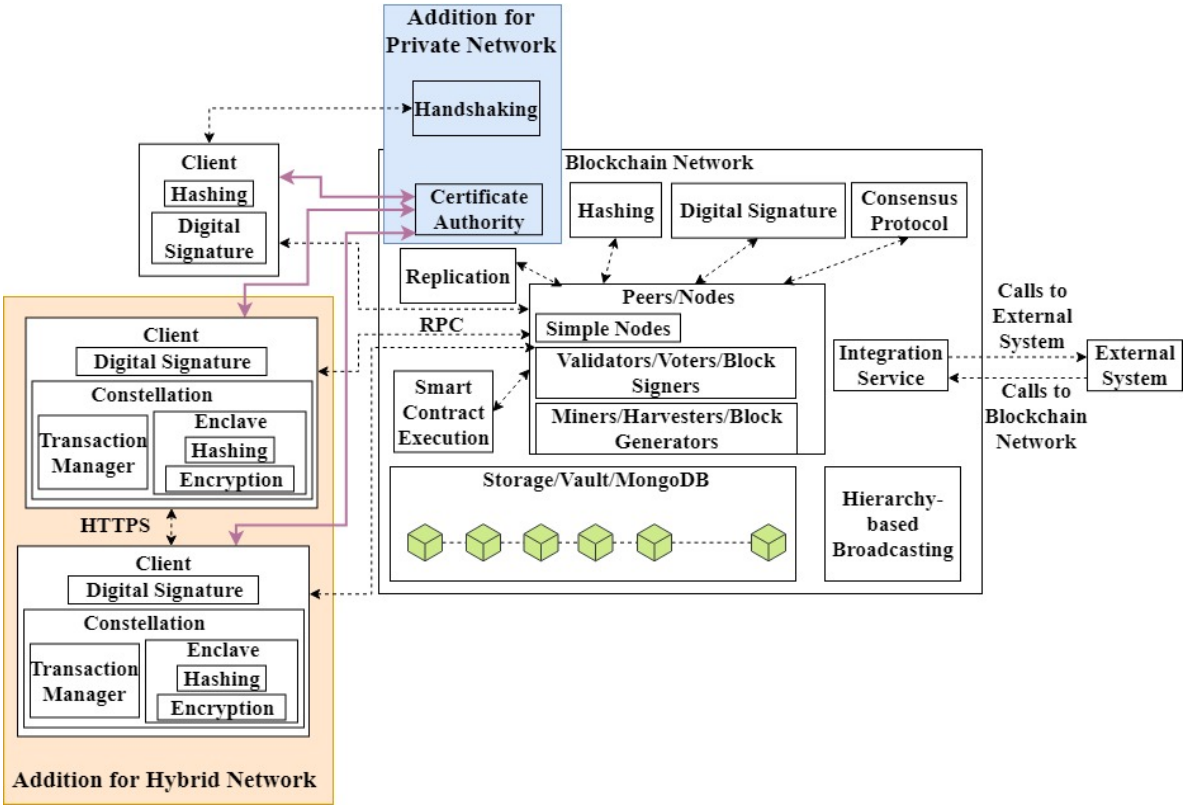

Figure 6. Single-ledger-based Blockchain Architecture for Public, Private, and Hybrid Networks.

This architecture can be used in applications domains, such as transportation, supply chain and project management, digital content ownership, finance, and energy trading. Because the architecture allows anyone to join the network, the transaction data is public, and there is no access control, it is not suitable for private applications, such as healthcare, education, governance, and national policy.

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 in the public network architecture. A certificate authority and a handshaking mechanism are added as shown in Figure 6. The certificate authority provides authentication and authorization for users to join the network. The access-control mechanism defines the ledger queries and updates roles for each participant. The handshaking mechanism ensures the authenticity of the nodes participating in a transaction and establishes connections between the nodes. In 2015, the blockchain platform Multichain [36] 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 by 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 [37], Chain core [38], HyperLedger Sawtooth [39], Hydrachain [40], Hyperledger Iroha [41], Burst [42], NEM [43], and BigchainDB [44].

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 [45] 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 receivers of the transaction.
2. The transaction manager of A stores the transaction payload in its local disk and sends the payload to its enclave for encryption.
3. Participant A's enclave encrypts the payload by 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 are sent to the transaction manager.
6. The transaction manager stores the encrypted payload and key by using the hash value as an index. The encrypted payload, the encrypted key by B's public key, and the hash value are transferred to the transaction manager of B securely via the HyperText Transfer Protocol Secure (HTTPS) [46]. Similarly, the encrypted payload, the encrypted key by C's public key, and the hash value are transferred to the transaction manager of C.
7. The transaction managers of B and C reply with an acknowledgement message to the transaction manager of A.
8. Participant A's transaction manager replaces the payload data in its local disk by the hash value and broadcasts the data to the network.
9. The network nodes which receive the hash value lookup for the received hash value in their respective transaction manager. The nodes which find the hash value pass it along with the encrypted payload and the key to their enclaves (B and C in this case).
10. The enclave decrypts the symmetric key by using the participant's private key and then decrypts the transaction payload by using the symmetric key. The decrypted payload is 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 the hash value.

In summary, single-ledger-based blockchain architecture for a 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 a trusted domain, i.e., an organization or a federation of organizations. The architecture for a hybrid network should be used by the public applications requiring confidential transactions between a subset of network participants. In a public or a hybrid network, anyone can be a full node and/or a mining one. With more nodes participating as full and/or mining ones, the number of messages transfers increases in the network, limiting the blockchain scalability. Comparing the number of encryption/decryption operations in a transaction flow, the private network has fewer operations than the public one 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 hash. The encryption/decryption operations are computationally complex and energy-hungry [47].

4.2. Multi-Ledger-Based Architecture for a Private Network

In 2016, blockchain platform Hyperledger Fabric [48,49] introduced a multi-ledger-based architecture for private network, as shown in Figure 7. The aim is to enable confidential and private transactions between a subgroup of participants within an organization or a federation. The architecture divides the blockchain network into channels to enable private transactions between the members of a channel [50]. To perform a private transaction between participants within a subgroup, the architecture uses a collection. This is because creating a channel within a channel is a CPU intensive process consuming high energy as compared to the creation of a collection within a channel [51]. The validators are known as peers and the miners are known as orderers in this architecture. There are two types of peers: 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 to be used for future references.
2. The transaction data is digitally signed and broadcasted to the network. The broadcasted transaction payload is called transaction proposal.
3. The transaction payload is validated by the endorsers. Each endorser executes the chain code associated with a valid transaction.
4. Each endorser digitally signs the transaction and sends back a proposal response to the client.
5. The client waits until a predefined number of proposal responses are received. The client aggregates all the responses and sends them with the transaction to the orderer.
6. The orderer generates a block of transactions (valid and invalid) received from the different clients and updates its ledger. It broadcasts the block to the peers in the network.
7. Each peer in the network verifies the validity of the transaction in the block. A peer creates a block, called a vblock, of valid transactions (i.e., a transaction which has a threshold of endorsers signatures). The invalid transactions are logged in a file for future reference but are not included in the block.
8. The client is notified of the successful execution of the transaction.

The multi-ledger-based architecture can be used in applications domains that involve several collaborating organizations which require confidentiality of transactions among different subgroups. For example, collaborating universities, banks or hospitals. This architecture is implemented by Oracle [52].

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

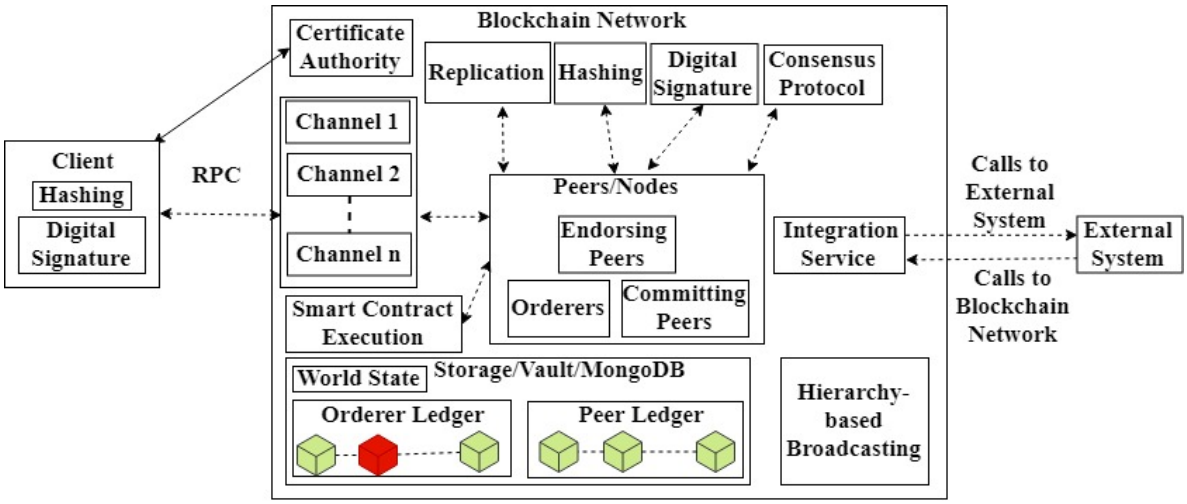

Figure 7. Multi-ledger-based Blockchain Architecture.

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, the transactions can be compressed as introduced by the blockchain development platform Credits [53]. They are compressed by using the lossless compression algorithm, also called Deflate; a combination of LZ77 [54] and Huffman algorithms [55]. In addition, the scalability issue must be addressed in a blockchain architecture caused by replicating the ledger among most of the network members, increasing the computational and communication overheads. As a research direction, scalability can be enhanced for instance by introducing a lightweight blockchain architecture that divides the network participants into clusters and selects a cluster head(s) for each cluster [56]. The cluster head(s) can be selected either based on voting or on the number of incident edges from a node [57]. The ledger can only be replicated on the cluster heads. The remaining participants can query the ledger to get the transactions information.

4.3. Interoperability-Based Architecture

The rapid adoption of blockchain by different applications domains has 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 interoperate between different blockchains. In 2017, the blockchain platform Elements [58] introduced the architecture for the interoperability between the public and the private blockchain networks, as shown in Figure 8. The architecture is also used to enhance the security of a blockchain by linking it to another blockchain. The platform Openchain [59] uses it to link its blockchain (let us call it sidechain) to the Bitcoin blockchain (let us call it mainchain). Whenever a block is added in the sidechain, a cumulative hash for that block is calculated by hashing the block hash with the previous block cumulative hash. The current cumulative hash is recorded in the block of the mainchain. This process is known as anchoring and it ensures high immutability of the sidechain.

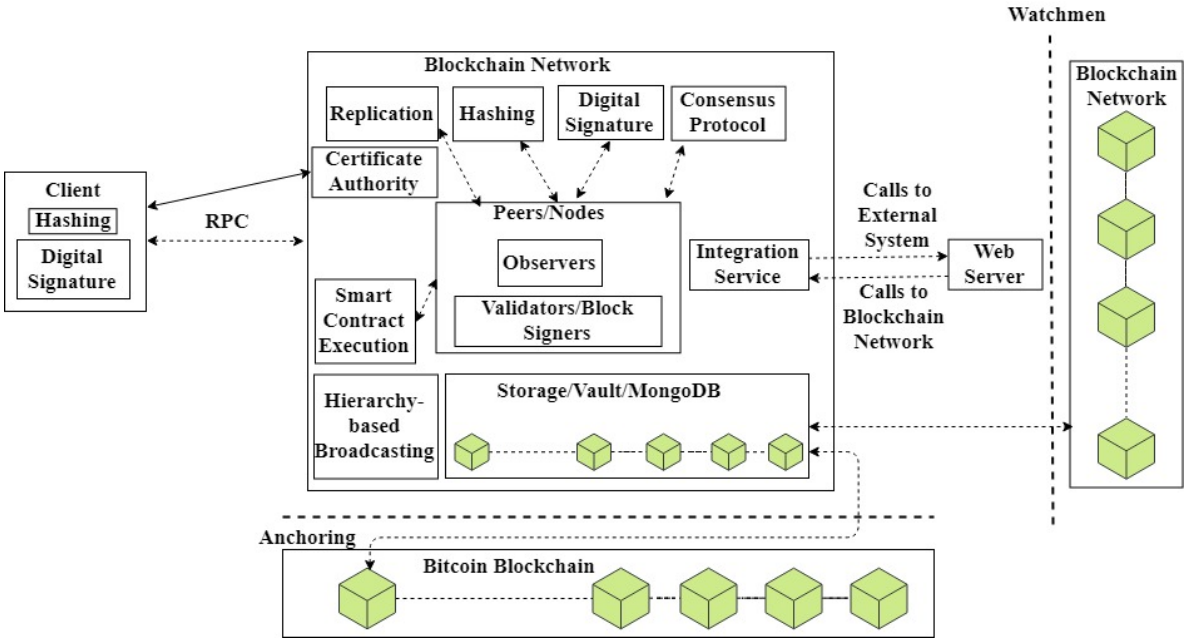

Figure 8. Interoperability-based Blockchain Architecture.

To perform an inter-blockchain transaction, the architecture uses a service called Watchmen. Watchmen is an authorized member which enables secure and verified inter-chain assets transfers. The architecture is developed by the Lisk platform [60]. The transaction execution flow in this architecture between two different blockchains is as follows:

1. A node issuing a transaction in the first blockchain creates and signs the transaction payload and sends it to Watchmen.
2. Watchmen validates the transaction and releases it to the second blockchain.
3. A block signer (similar to a miner) in the second blockchain adds this transaction in a block and the ledger is updated after the block was 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 blockchain architectures and the mapping of the different blockchain platforms to the classified architectures. It describes the specificity of each platform, the network type, the operating system(s) supported, and the programming language(s) to develop applications. It also states whether or not a platform supports application portability. The portability is ensured by generating bytecodes.

Table 2. Taxonomy of Blockchain Architectures.

Architecture Type	Supporting Platform	Platform Specificity	Network Type	Operating System(s) Supported	Programming Language(s)	Applications Portability	Applications
Single-ledger-based	Ethereum [34]	-	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 [37]	-	Private	Linux	Go	Yes	Education, healthcare, governance, national policy, and research
	Chain core [38]	Validator is known as block signer and miner is known as block generator		Linux, Windows and Mac OS	Ivy		
	HyperLedger Sawtooth [39]	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 [40]	The role of miner is performed by a validator		Ubuntu	Python	No	
	Hyperledger Iroha [41]	Validator is known as peer and miner is known as orderer		Linux and Mac OS	Python, Java, JavaScript, C++		
	Burst [42]	-		Linux, Windows and Mac OS	JavaScript		
	NEM [43]	Miner is known as harvester		Linux, Windows and Mac OS	Java		
	BigchainDB [44]	Validator is known as mongodb server and the role of miner is performed by a validator		Ubuntu	JavaScript, Python		
	MultiChain [36]	The role of miner is performed by a validator		Linux and Windows	Python, C++, JavaScript, Ruby, Go		
	Quorum [61]	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 [53]	Validator is known as trusted node and miner is known as		Linux and Windows	Java		

Table 2. *Cont.*

Architecture Type	Supporting Platform	Platform Specificity	Network Type	Operating System(s) Supported	Programming Language(s)	Applications Portability	Applications
Multi-ledger-based	Hyperledger Fabric [48]	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 [52]			Linux, Windows and Mac OS	Java		
Interoperability-based	Elements [58]	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 [60]	-		Ubuntu 14.04.5, Windows and Mac OS	JavaScript		
	Openchain [59]	The role of a miner is performed by a validator and a simple node is known as observer		Linux, Windows and Mac OS	JavaScript		

5. Evolution of Consensus Protocols in Blockchain

A transaction in the blockchain is considered valid after the network participants have reached a consensus using a consensus algorithm. However, there is a gap between the existing consensus algorithms and the application 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 them based on compute-intensive, non-computing capabilities, and voting. 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. In this section, we present the different compute-intensive consensus protocols used in the blockchain literature.

5.1.1. Pure Proof of Work (PoW)

PoW was first introduced by Dwork and Naor in 1992 [62] and used by Back A. in 2002 [63] to reduce the number of spam emails by making it computationally complex and time-consuming to send multiple emails simultaneously. In [62], a pricing function should be calculated before sending an email. In [63], a hashcash function which includes a counter value should be calculated in a way that the hashed output has a predefined number of leading consecutive zeros. The counter value is adjusted by using the brute-force [64] method. The hash function used in hashcash is SHA-1 [65]. The obtained hash and the counter value should be added to the email header before sending the email. The recipient can regenerate the hash for verification [66].

The PoW algorithm was used in 2008 by Nakamoto et al. [2] in the Bitcoin blockchain network. The blockchain mining nodes compete against each other to generate a valid block of transactions. A mining node should hash the block data by using a counter, as in hashcash, in addition to the requirement that the hash output should be below a particular threshold. This increases the computational complexity of mining compared to hashcash. In blockchain, the counter value is known as nonce. To calculate the block hash, a mining node hashes the Merkle root hash value, the timestamp, the previous block hash, the block version, and the nonce value. The hash function used by the Bitcoin blockchain is SHA-256 [67]. The miner who obtains the desired hash value, adds the nonce 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. A valid block is updated in the ledger and the miners start mining the next block. The miner whose valid block is stored in the ledger receives a mining incentive for the computational power used. 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 given to the miner by the entire network [68].

The requirements on the block hash value determine blockchain PoW algorithm difficulty level which is dynamically adjusted to maintain a constant block generation interval. The difficulty level of the network increases exponentially with increasing number of zeros. In the Bitcoin network, it is adjusted after every 2016 blocks to maintain the average block interval at 10 min [69]. The change is calculated based on the ratio of the ideal time to generate 2016 blocks, i.e., 20160 min (1 block every 10 min) and the actual time required to generate the last 2016 blocks. The target threshold hash value for a block for the next 2016 blocks to be mined is calculated by using the current target value as the reciprocal of the difficulty. Equations (1) and (2) show the calculation of the difficulty and the target value in the network, respectively.

$$D_{new} = D_{current} \times \frac{20160}{T_{2016}} \quad (1)$$

$$Target_{new} = Target_{current} \times \frac{T_{2016}}{20160} \quad (2)$$

where D_{new} and $D_{current}$ are the new and current difficulty levels respectively, and $Target_{new}$ and $Target_{current}$ are the new and current target threshold levels respectively.

The Bitcoin network implementing PoW has a throughput of 60 transactions per second [70]. PoW 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 can form 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 [71]. 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 the users [72]. This is known as the problem of 51% attack in PoW. Moreover, PoW suffers from possible security attacks such as routing, sybil, eclipse, time jacking, and bribery attacks as discussed by Conti et al. [73]. In addition, 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. Furthermore, the mining competition to win the incentives exacerbates the energy consumption problem. This is because of the high-performance hardware resources used for brute-forcing the nonce value. In 2017, the Bitcoin mining used around 30.14 TerraWatt hours (TWh) of energy which is equivalent to the energy usage of entire Ireland in a year [3]. The annual energy consumption as of 22 June 2019 is 67.937 TWh [74]. This increasing amount of energy consumption is affecting the environment with increasing global warming. 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 countries such as Sri Lanka and Jordan [4]. The PoW consensus is implemented in Bitcoin, Litecoin [75], and Dogecoin [76] networks.

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

In 2013, King S. proposed prime number PoW to channelize the high amount of energy consumption of PoW for dual use [77]. Similar to the calculation of nonce in PoW, Prime number PoW involves a computational calculation of Cunningham chain of prime numbers [78] that can be 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 the development of cryptography methods. The computed prime chains are published in the ledger. The chain of the prime numbers must satisfy two requirements: (1) it should be a Cunningham chain of either first kind [80], second kind [80] or bi-twin [81], and (2) its length should be larger than a target length.

To ensure that a prime chain is calculated 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 becomes proof of work certificate. The certificate is hashed along with the block header to calculate the block hash. The validity of the block is verified by checking whether the block hash is correct, and the prime chain is valid, and its length is above a target length. In order to check the primality of the prime chain, the miners use the classical Fermat test [82] together with Euler–Lagrange–Lifchitz test [83]. The miner whose block is updated in the ledger receives an incentive that includes the mining fee and the transaction 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 $p_1, p_2, \ldots, p_k$ of length k, the difficulty for the next prime chain is then calculated by using Equation (3).

$$d = k + \frac{p_k - r}{p_k} \quad (3)$$

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

The prime number PoW consumes a high amount 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 [77]. 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 is not evaluated in terms of performance and protection against security threats. The prime number PoW is only used by the cryptocurrency primecoin [84].

5.1.3. Delayed Proof of Work (DPoW)

The PoW is an energy-hungry consensus protocol used by the Bitcoin network for security. Delayed Proof of Work (DPoW) proposes to use this compute-intensive security of PoW to secure other blockchain networks that use an energy-efficient consensus protocol. Hence, DPoW is a hybrid consensus method that secures a blockchain network using the mining power of PoW blockchain. In DPoW, a group of 64 notary nodes (elected by the stakeholders in the network) are responsible for generating a block. Each notary node validates the transactions and creates a block in a round-robin fashion without involving the compute-intensive and energy-hungry calculation of the mining proof. However, to ensure the 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 [85] which uses the mining power of Bitcoin blockchain. DPoW solving the issue of high energy consumption of PoW and prime number PoW compromises the network security in a situation when the communication between the DPoW and the PoW blockchain is interrupted or lost. Moreover, the consensus protocol is not evaluated 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 broadcast 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). All the nodes in the network maintain a copy of the blockchain ledger (let us call it the mainchain). A node, which receives block A, validates it and appends it to its copy of the mainchain and a node, which receives block B, validates it and appends it to its copy of the mainchain. When a node with block A in its mainchain receives block B, the node verifies the validity of block B. However, both blocks A and B have the same parent block. Consequently, the node initiates a new chain (let us call it the secondary chain) separate from the mainchain and appends the block B in that secondary chain. Similarly, a node with block B in its mainchain appends block A in its secondary chain after it has been received and validated. In the blockchain, when the chain is divided into two parts in this way, it is known as forking [25]. 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 mainchain will be considered valid as it becomes long compared to one with block B in the mainchain. Similarly, if the new block contains block B as the parent block, the blockchain with block B in the mainchain will be considered valid. The blocks in the side chain are then removed and the chain continues to develop on the mainchain. Generally, the blockchain fork is resolved within one block [25]. Therefore, to handle transactions in the forked block, all the transactions in a block are executed after a certain number of blocks are appended to the longest chain. For instance, in the Bitcoin network, it takes 6 blocks confirmation for a transaction to get executed [86].

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 where all the miners use their computing power to win the right to mine the next block. 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 consensuses under capability-based protocols. Different capability-based protocols are explained below.

5.2.1. Proof of Stake (PoS)

Proof of Stake (PoS) was proposed in 2011 [87] and used by the cryptocurrency Peercoin (also known as PPcoin) in 2012 [88]. The miners in PoS are called forgers and the mining process is known as forging. At the beginning of a forging round, each forger deposits a certain amount of owned cryptocurrency coins in the network as a 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 [88], a forger with 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 a minimum of 30 days. The stake-holding duration is involved to avoid repetitive selection of a forger with a greater 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 [89], a forger with 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 a private key in the calculation generates a unique hit value for each forger in the network. The forger with the hit value below a target value is selected for the process of forging. The target value is calculated by using Equation (5). To make the selection based on the capability of the miner, the calculation of the target value involves the number 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 with 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 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 with a high value of cumulative difficulty is selected. The cumulative difficulty mentioned is calculated as stated in Equation (6) [89]. The forger who forges the block receives the transaction 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 like 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 to them as compared to PoW where the computational power used by the miner is not 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 [90]. This is by selecting the forgers based on election rather than on the amount of staked coins owned. In DPoS a group of nodes known as witnesses (also called 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 vote is proportional to the number of coins that the node owns. A node can vote multiple witnesses with a single vote for each witness. The first N witnesses with the highest votes are then selected for the mining process to avoid a single witness from 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 most of the nodes have voted for at least 10 witnesses, then the first 10 witnesses with 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. 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 [91], Nano [92], and Cardano [93]. 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 [94] 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 the linear function used by PoS (Equation 4). Consequently, the growth rate of the coin age follows an exponential decaying line [94]. 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 [94]. 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 [95].

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 [96]. In PoB, the miners need to burn the coins by sending them to an 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 cannot be further used. This discourages the malicious miners from mining an invalid block as a miner will spend coins 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 decays exponentially with time. The transactions performed for sending coins to the eater 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 by using Equation (7) [97].

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

The internal hash is calculated by hashing together the burned transaction 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 continued 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 by using Equation (8) [96].

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

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

PoB is used by cryptocurrency Slimcoin [98] and third generation coin [99]. 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 previously discussed protocols in addition to the issue of high energy consumption in the computation-based protocols, Dziembowski et al. proposed proof of space (PoSpace) also known as proof of capacity in 2015 where a miner with enough disk space wins the right to generate the next block in the chain [100]. 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 [101] 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 the 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 to 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 a scoop, and for 8192 hashes there exist 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 to be 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 the next block. A miner in PoSpace is known as a 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 previously discussed capability-based protocols. Moreover, the protocol does not require any specialized 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 to mine the block, encouraging malicious users to generate an invalid block. PoSpace is currently used by cryptocurrency such as Spacecoin [102], Chia [103], and Burstcoin [104].

5.2.6. Proof of History (PoH)

Proof of History (PoH) proposed in 2017 by Yakovenko aims to address the issues in compute-intensive-based and above discussed capability-based protocols [105]. In PoH, the validators are referred to as the verifiers and the mining nodes are referred to 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 a 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 cannot be

parallelized on a multi-core architecture as the output of the function cannot be known beforehand. On the other hand, the proof can be verified in parallel by the verifiers on a multi-core architecture making it less time-consuming. Equations (9) and 10 show the time taken for the hash generation and verification respectively on the same multi-core machine [105].

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

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

In the process of verifying the proof generated by the leader, each verifier signs the proof and send it back to the network. If the 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 number of coins staked by each verifier.

As PoH does 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 nowadays) to increase the speed of verification. In addition, as the hashing function is continuously 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 [43] addresses the issue of reduced transaction flow existing in the PoS protocol where the miners do not perform transactions to increase their chances of mining. In PoI, the miners are referred to as harvesters and the process of mining is known as harvesting. A miner with 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 number 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 number of vested tokens, which we call vesting amount. 10% of the vesting amount vests each day [106], 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 importance score. PoI rewards miners in terms of importance score upon performing transactions with the nodes who have 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 advantage by transacting back and forth [106]. Finally, the importance score also depends on the number of transactions performed by the miner in the 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 by using Equation (11) [43]. 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 occurred. The exponential decay function in 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} [43].

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 among 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 in PoS [107]. 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 [107] 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 includes providing storage space, dedicating computing resource, and reviewing third-party applications. The servi tokens cannot 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 subgroups randomly and each subgroup 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 the 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) [108] 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 [109,110]. The role of a miner in PoAuthority is performed 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-checking. In order to be a validator, the authority must 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 [111] 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 [112]. 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 [112] and Menlo one [113]. 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)

To solve the issues of rich getting richer and centralization of the network, Proof of Elapsed Time (PoET) was developed by Intel in 2016 [114] as a cost-efficient consensus protocol. PoET uses a

Trusted Execution Environment (TEE) [115] and Intel's Software Guard eXtensions (SGX) [116] 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 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 waiting time by using the code executing in TEE and then sleeps during that period of time. The calculation of the random waiting time is done by using the formula stated in Equation (12) [117].

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

where MinimumWait is a fixed system parameter, LocalAverageWait is calculated by using the number of active nodes in the network, and $r \in [0, 1]$ is a real number which is derived from the hash value of the node's previous signed attestation. The more the number of active nodes in the network is, the more will be the waiting time of the nodes to avoid collision [117].

The first node waking up propagates a signed certificate to the network indicating that it has been selected as the leader. The remaining network nodes check that 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 by using Equation (13) [117].

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

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

The leader creating a new block receives transactions fee. PoET has low energy consumption compared to PoW and does not favor 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 the controlling authority leading to a less decentralized blockchain. In addition, PoET is vulnerable to malicious attacks [118]. PoET is currently used by the HyperLedger Sawtooth blockchain platform [39].

5.2.11. Proof of Activity (PoA)

In pure PoW, the miners receive a mining fee in addition to the transactions fees to encourage the miners to participate in the process of mining and therefore securing the network. The current mining reward is 12.5 bitcoins and it is halved after every 210,000 blocks mined. The PoW becomes less significant when the mining reward is obsolete, and the miners only rely on the transactions fee. Consequently, the miners dedicating their computing power may demand high a transaction fee discouraging the use of the network. This is because with low transactions fees the cost of mining will be much higher than the incentive received. This issue is known as the tragedy of commons [119], where everyone selfishly looks for own benefit without contributing to the network security. In addition, in most of the consensus protocols, the validators of the transactions receive no reward for their work. Proof of Authority (PoA) [120] developed in 2014, address these issues by using combining PoW and PoS algorithms in one. This is achieved by dividing the transactions fees between the 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 to generate a block with a particular nonce. However, a miner in PoA generates an empty block. The miner broadcasts the

created block to the network. In the second phase, PoA selects N validators referred to as stakeholders based on the number of coins they have by using the PoS algorithm. Each selected stakeholder verifies and signs the block, and broadcasts it into the network. The block is 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 hashes the block and broadcasts it to the network. The transactions fees for the transactions included by the N^{th} stakeholder are shared between the miner who created the block and the N stakeholders.

PoA suffers from the issue of high energy consumption as in PoW and it favors the rich as in PoS. PoA was not tested in terms of performance and security threats. It is used by the cryptocurrency Decred [121].

In summary, the capability-based consensus protocols reduce the high energy consumption of the compute-intensive-based protocols but suffer from the issues of the rich getting richer, the encouragement of malicious activities, and the 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 use a voting system to elect a miner for generating a block. They eliminate the issue of high energy consumption of compute-intensive-based protocols due to the miner selection based on a competitive approach. They also address the problem of the rich getting richer in capability-based protocols as the selection is based on 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 may behave maliciously. In reference to distributed systems, byzantine fault tolerance is the ability of the network to reach the desired consensus despite some of the nodes in the system are failing or behaving maliciously [122]. 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 [123], a distributed computing network term for a situation where the network nodes must agree on a single state to avoid 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 1999 [124]. In PBFT protocol, a group of nodes is selected by a central authority with one node as the leader and the others as the backup nodes. All the nodes in the system communicate with each other with the aim of reaching an agreement by assuming that all the honest nodes have the exact same copy of the ledger. For the PBFT protocol to function correctly, the number of malicious or crashed nodes must less than $\frac{n}{3}$ (n is the number of nodes in the system). The network is more secure with increasing number of nodes as it is more unlikely to have $\frac{n}{3}$ of the network being malicious.

Each round of generating a block in PBFT is known as a view and can be broken down into four phases in the context of blockchain. (1) A client sends a request to the leader node to perform a transaction. (2) The leader node collects the transaction requests and groups them in a block. The block is then broadcasted to the backup nodes. (3) Each backup node verifies the transactions in the block and creates a block of valid transactions. The node 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 added to the ledger of that node.

PBFT is used by the development platforms Hyperledger Fabric, Hyperledger Iroha, Oracle, Hydrachain, and BigchainDB. In Fabric and Oracle, the leader is referred to as the orderer and the backup nodes are known as the peers. In BigchainDB, PBFT is called tendermint protocol [125] where the nodes with a 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 than PoW [126]. However, PBFT needs an authority service for the selection of the leader and the backup nodes, making it less decentralized. For a private network with 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 scalability becomes an issue due to the communication overhead. In addition, this communication overhead may lead to an increase in energy consumption. 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 [127] in a public network.

5.3.2. Delegated Byzantine Fault Tolerance (DBFT)

To avoid the centralized selection of a leader and backup nodes in PBFT, the DBFT protocol was proposed by NEO cryptocurrency platform [128] in 2014. In this protocol, the nodes are selected by using a voting process. A leader node is known as a speaker and the backup nodes are referred to as the delegates. The selection of the delegates is done by using a voting system, where the network participants holding cryptocurrency participate 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 is 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 this case, where all the participants are selected as delegates, the network suffers from the issue of communication overhead. For DBFT to function correctly, the total number of malicious or failing nodes in the network should be less than $\frac{(2n-1)}{3}$ out of the total n nodes in the network. The issues of communication overhead and sybil attacks still exist in DBFT.

5.3.3. Federated Byzantine Agreement (FBA)

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. [129] as a completely decentralized version of PBFT used by the Ripple network [130]. Compared to PBFT, FBA does not require a list of selected nodes by a central authority to validate and process the transactions. FBA has an open membership similar to PoW, where any node in the network can participate in the consensus process. A new transaction is added into the network if 80% of the nodes agree on the status of the transaction (which is 66% in PBFT). With all the nodes participating in the consensus and broadcasting the transaction status to each other to get 80% of the confirmation, the network will suffer from communication overhead. To address this issue each node in the network communicates with a list of nodes, known as a Unique Node List (UNL) [18]. According to the ripple network, the intersection of UNL by any two nodes in different UNLs should be at least one fifth of the total nodes in the network. This ensures that a transaction is validated by all the nodes in the network. When receiving a transaction, a UNL node validates the transaction and updates a candidate list. Each node broadcasts its candidate list to all the nodes in the UNL. A transaction is considered valid if 80% of the nodes confirm that it is valid. For FBA to function correctly, the total number of malicious or failing nodes in the network should be less than $\frac{(n-1)}{5}$ (n is the total number of nodes) [129]. FBA is also used by Stellar cryptocurrency platform where the UNL is called quorum slices [131]. In stellar, all the nodes in a quorum slice must agree on a transaction to update the ledger, unlike ripple where only 80% of the agreement is 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 [53] in 2018, uses DPoS consensus algorithm for the selection of the nodes participating in the consensus process, and the BFT algorithm to update the ledger while protecting against malicious attacks. The algorithm of DPoS+PBFT can be divided into two phases: (1) the selection of nodes and (2) the ledger update.

In the first phase, the algorithm selects the head nodes and the trusted nodes to participate in the consensus process. The head nodes provide the confirmation of the transactions and create a block, while the trusted nodes create a list of valid transactions to add in the block using BFT. At each round of block creation, all the nodes in the network can participate in the selection as a head node or a trusted node. In order to get selected, a node should send the hash of the last block in the ledger to the previous head node who created it in a predefined period of time. After this time has elapsed the nodes which were unable to send the hash value are eliminated from the participation. In addition, the previous head node eliminates the nodes which send the wrong hash value 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 selected as the head node and a certain number of remaining nodes from the list are selected as the trusted nodes. The number of trusted nodes is calculated by 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 is the total number of eligible nodes in the list. The list containing the head node and the trusted nodes is broadcasted to all the nodes in the network.

In the second phase, the transactions proposed while the first phase is running are sent to the newly selected head node by all the nodes. The head node generates a list of transactions and sends it to all the trusted nodes. The trusted nodes validate each transaction and create a list of valid transactions. Each trusted node sends its list of valid transactions to all other trusted nodes. Each trusted node creates a list of approved transactions by selecting the transactions with most validations (using BFT) in all the lists received from each trusted node. Each node sends its list of approved transactions to the other trusted nodes. A node with a mismatching list is considered to be faulty or malicious, and it is eliminated. The list is sent to the header node which creates the block of transactions in the list. The node broadcasts this block to the other nodes in the network for verification. A new round begins with the first phase. The DPoS+BFT protocol increases the communication overhead tremendously compared to the PBFT because the amount of data transfer between the trusted nodes is twice more than the amount of data transfers between the backup nodes in PBFT. In addition, the amount of data transfers between the head nodes and the trusted nodes in DPoS+BFT is more than between the leader and the backup nodes in PBFT. Consequently, DPoS+BFT consumes more energy than PBFT.

5.3.5. Raft

The issue of communication overhead 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 [132]. The algorithm works correctly as far as more than 50% of the network nodes are working normally. Each node in the network is in one of the following states: 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 take place by using RPCs. AppendEntries RPCs are initiated by the leader for the replication of log and RequestVote RPCs are initiated by the candidates during the election. In the candidate state, the node elects a new leader. The algorithm 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 arbitrary length known as terms. Each term is consecutively indexed and the election for the leader takes place at the beginning of a term. Initially, a node is in a follower state and continues to receive AppendEntries RPCs from a previous leader. If a node stops receiving the RPC requests from the leader for a specified period of time known as election timeout, the node assumes that there is no leader in the network and begins a new election. To begin a new election, the follower node increments the index of the current term, changes its status to a candidate state, and participates in the election. The candidate votes for itself to be the new leader and simultaneously issues RequestVote RPCs in parallel to all the other nodes. All the other candidates respond to the request by submitting their votes. This is continued until one of the possible outcomes occurs: (i) the candidate wins the election, (ii) another candidate wins the election, or (iii) no candidate is elected in the term.

  (i) If the candidate receives the votes from the majority of the nodes in the network, then the candidate wins the election and sends a message to all the other nodes stating its authority as the leader.
  (ii) While waiting for the votes, a candidate may receive an authority message from 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 rejects the message. If the index is more than that of the candidate, the candidate recognizes the node as the leader and changes its state back to a follower.
  (iii) If multiple followers begin the election process simultaneously, the votes may split in a no candidate receives the majority of votes. In this case, each candidate times out and starts a new election process by incrementing the term index. To avoid the split of votes 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*: The elected leader 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 transactions. The leader creates a block of these transactions and issues an AppendEntries request in parallel to all the followers. The block is replicated by each follower, and an acknowledgement is sent to the leader to confirm the replication.
- *Transaction execution*: When the leader receives the majority of the acknowledgements from 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 continues until the term ends.

Raft consensus algorithm solves the issue of crash tolerance but does not provide data integrity in case a node behaves maliciously. If the leader node behaves maliciously and executes invalid transactions, the entire blockchain becomes invalid. If most of the followers behave maliciously and do not send the replication acknowledgements, the block cannot be committed by the leader and the network suffers from the issue of high latency. The raft consensus algorithm is used by the blockchain development platform Quorum [61].

5.3.6. Federated

Federated CFT-based consensus is proposed by the blockchain development platform chain core [133] where the leader and the backup nodes are elected from a group of authorized nodes.

The leader known as the block generator is responsible for transactions validation and the 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 validate all the transactions. The block generator 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 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 updates its ledger and broadcasts the block to the network. All the network nodes verify that the block has enough signatures and update their own ledgers. If the block does not contain the required number of signatures, a new block generator is elected. The federated protocol makes the network less decentralized as the selection of the leader and the backup nodes is performed by a centralized authority.

In summary, the compute-intensive-based consensus protocols suffer from the issues of high energy consumption, environmental pollution, low transaction throughput and low scalability, whereas the capability-based protocols solve the issue of high energy consumption but tend to be biased towards the rich (wealth dominance) and more prone to malicious attacks. The voting-based protocols solve the issues of high computational energy consumption, low transaction throughput and scalability in the compute-intensive-based protocols but they make the network less decentralized. Moreover, the number of data transfers is high in voting-based protocols leading to higher energy consumption. Consequently, there exists a need for energy-efficient, decentralized, high transaction throughput, and highly scalable blockchain consensus protocol to address the misalignment between the existing protocols and the customer services where applications are evolving rapidly to meet the requirements of a collaborative large-scale ecosystem. One possible solution is in the direction proposed by Tromp J. that uses cuckoo cycle-based PoW [134], 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 and providing two possible locations for each key. The mining in the cuckoo cycle involves finding a cuckoo graph cycle of a specific length, as denoted by the network difficulty level. In addition, very few consensus protocols discussed in this paper were evaluated in terms of performance and security threats. Moreover, the works in the literature which evaluated some protocols in terms of performance and security threats used different experimental environment and setup making it difficult to conclude an objective comparison between the protocols. To our knowledge, there is no work in the literature comparing the performance evaluation of all the consensus algorithms in a unified experimental environment and setup.

6. Service Creation and Innovation Capabilities

Blockchain initially designed for Bitcoin has shown its adaptability in various sectors beyond finance and banking without the need for a centralized trusted third party. The inherent characteristics of the technology such as local data access, fault tolerance, immutability, privacy, authenticity, and security have attracted government and private organizations for its adoption. In this section, we provide a taxonomy of several existing blockchain applications and 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 the application domains. It also shows the type of blockchain network (public, private or hybrid) used by these applications.

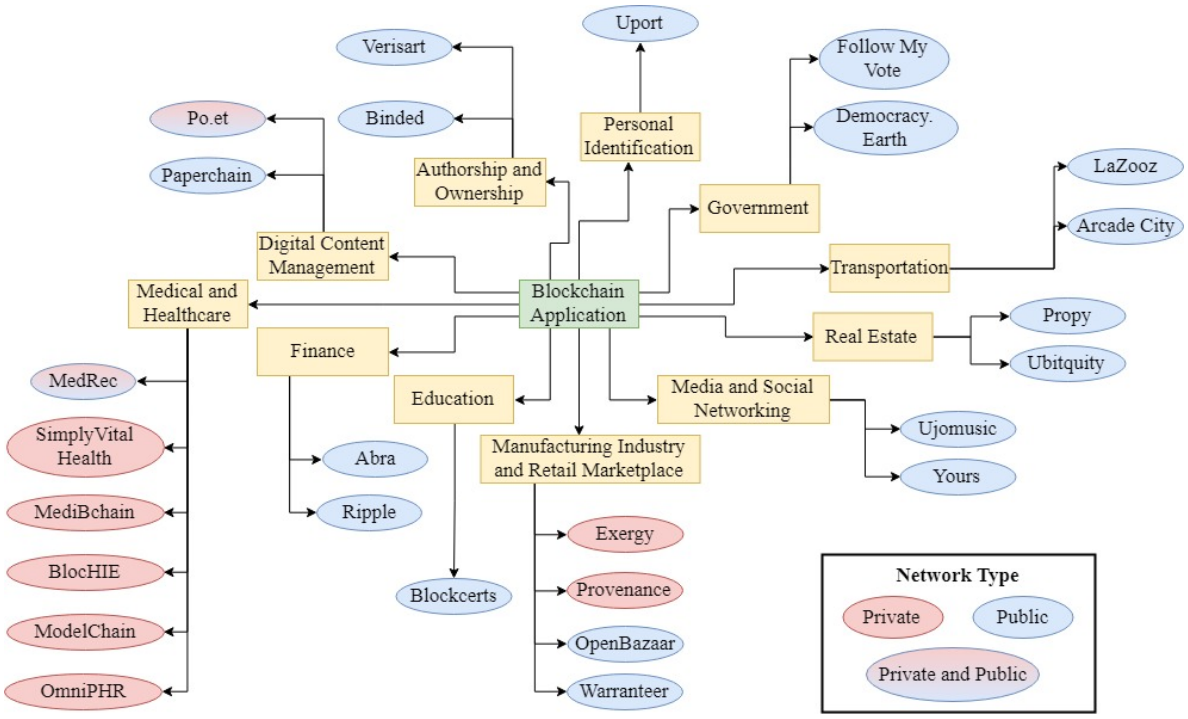

Figure 9. Existing Blockchain Applications in Different Domains.

6.1.1. Education

- *Blockcerts*: The application allows educational institutions to create and issue digital student certificates over the blockchain network. The students and the employees can view the certificates [135,136]. The application makes each individual capable of owning and sharing his/her own digital certificate on a peer-to-peer network. [137] develops a similar application and analyzes its performance within a University network.

6.1.2. Medical and Healthcare

- *MedRec*: The management of Electronic Health Records (EHRs) has become complex due to scattered medical data of patients moving across different medical organizations. Consequently, it becomes difficult to track patient medical history. MedRec provides easy access and management of EHRs using the blockchain technology [138]. It does not use the ledger for updating the record but encodes the metadata containing the information about ownership, permission and the integrity of the data so that records can be accessed securely.
- *SimplyVital Health*: Coordinating a patient's treatment between a team of therapists, practitioners, and specialists is complex when the patient is receiving treatment from multiple healthcare providers. It is difficult to get analytics and insights from the patient data to support better treatment. SimplyVital Health allows different medical experts and organizations to share patient medical data securely over a private blockchain network [139]. The ledger is maintained by the patient's healthcare providers. The platform uses machine learning algorithms to forecast the 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 that allows patients to update and share their personal health data [140]. The security of the data and the privacy of the patient are achieved by the blockchain characteristics. To tackle the issue of breaching, the data is encrypted before updating the blockchain ledger.

- *ModelChain*: A blockchain framework to preserve the privacy of medical health data by using machine learning techniques. It involves a scalable proof of information algorithm to determine the order of the online learning process [141].
- *OmniPHR*: A blockchain healthcare data management architecture to manage the scattered personal health data of patients [142]. The architecture allows patients to update their data using blockchain to have a unified view via access-control rights.
- *BlocHIE*: A blockchain platform to store and share Electronic Medical Records (EMRs) and Personal Health Data (PHD) [143]. EMRs include the medical reports and diagnostic results prepared for a patient by a health professional whereas, PHD includes the patient data generated from sensors and health monitors. The platform uses two different blockchains to store EMRs and PHD. To reduce the communication overhead, the medical data is stored in a local database by the doctors and the patients and the hash of the data is stored in the blockchain ledger to ensure the privacy and authenticity of the data.

6.1.3. Finance

- *Abra*: A mobile application that allows users to buy, sell, invest, and trade different digital and fiat currencies [144]. Abra resides in the Bitcoin network, with the miners and the validators maintaining a 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 application for currency exchange and remittance for global payment and real-time transaction tracking [130]. Ripple connects different banks and payment providers over the blockchain network for transactions. It allows users to trade assets among each other by using the ripple native currency.

6.1.4. Manufacturing Industry and Retail Marketplace

- *Exergy*: A private blockchain platform that creates localized energy marketplaces for trading energy across existing grid infrastructures [145]. Exergy allows prosumers to transact the energy they generate using renewable resources in near real time with the consumers on the network creating a more efficient and sustainable community. The participants verify each transaction and maintain a copy of the ledger.
- *Provenance*: A blockchain platform to help business in developing trust by enhancing immutable and secure supply chain [146]. The network involves the consumers, producers, manufacturers, certifiers and auditors, and registrars. The consumers maintain a copy of the ledger. The consumers can trace and verify the origin, attributes and impact of the product.
- *OpenBazaar*: A peer-to-peer e-commerce that allows users to create their online store and to sell their products using cryptocurrency [147]. OpenBazaar is a Bitcoin powered marketplace where the users and buyers can trade directly eliminating the third-party fees. No banks or financial organizations are involved as the application does not use fiat currency.
- *Warranteer*: An application that allows users to save the warranty of their products electronically in a blockchain network [148]. The users upload the warranty of a product using the barcode and get notified for warranty status and expiration. The customers can get direct services from the service providers in case of a product malfunction by interacting with them over the network. Warranteer stores the data in the cloud over the Bitcoin network.

6.1.5. Media and Social Networking

- *Ujomusic*: A blockchain framework that allows music creators to own the rights of the music they create [149]. A music creator can directly trade a music content with the fans using cryptocurrency.

Ujomusic runs over the Ethereum network and has an embedded wallet to pay for the music content in Ethereum's digital currency.

- *Yours*: A social networking application to share valuable information with the community [150]. The application runs in the Bitcoin network and has an embedded Bitcoin wallet to pay the content creators and curators for their contributions.

6.1.6. Real Estate

- *Ubitquity*: A blockchain platform that allows transparent recording and tracking of title and property in the ledger [151]. It involves the property owners, the buyers, the real estate industry, and the government authorities in the network.
- *Propy*: A blockchain platform for overseas property transactions. The users can sell/buy a property online from any country by using either fiat currency or cryptocurrency [152]. All the documents related to a property transaction are added to the Ethereum network.

6.1.7. Transportation

- *Arcade City*: An application by which drivers can make local associations for ride-sharing and car-hiring underpinned by the Ethereum platform eliminating the need for a centralized cab service [153]. This would 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 using vehicles' unused space to create a variety of smart transportation solutions [154]. LaZooz synchronizes 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 blockchain voting system that connects all the voters and authorities [155]. This leads to a transparent election process making it traceable.
- *Democracy Earth*: An open source blockchain platform to develop decision-making smart contracts for democratic affairs at the state, the country, or the 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 identities, running on the Ethereum platform [157]. The application allows users to request and send credentials, and to safely store their keys and data, and sign transactions.

6.1.10. Authorship and Ownership

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

6.1.11. Digital Content Management

- *Paperchain*: A blockchain decentralized marketplace for organizations, content creators, and media companies to monetize their contents and to make them globally available [160].
- *Po.et*: A blockchain application to record a timestamped immutable information about content creation [161]. The publishers and content creators use the application to prevent content theft.

6.2. Emerging Blockchain Applications

There are many other application domains that are emerging based on the blockchain technology. We summarize these domains as follow:

6.2.1. Space

Blockchain can be used for improved data management and space communications in various interplanetary space missions, thanks to its characteristics. Recently, the National Aeronautics and Space Administration (NASA) awarded a grant to the University of Akron at Ohio state in the USA to develop a blockchain-based autonomous spacecraft system called Resilient Networking and Computing Paradigm (RNCP) as a first step towards a blockchain adoption [162]. Ethereum allowing the development of smart contracts can be used for building 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 blockchain-based space program known as coral [164] to facilitate the 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 by 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 a pilot program for the standard chartered bank to create a smart contract-based insurance policy for faster cross-border policy creation and execution [167]. The real-time system would allow different entities such as companies, their units, and the insurers to simultaneously share all data and documents related to policy in a transparent way.

6.2.3. Education

Blockchain technology has various applications in the sector of education and learning at the individual, institutional, group, national and international levels. Blockchain can be used to securely store digital records of the students [137,168,169], infrastructure security, transportation management for staff and students, smart contracts for the staff payments and students credits [170,171], human resources [172], data storing and sharing via libraries [173], research articles submissions, verifications and reviewing [174,175], inter-organizational data collections and analysis [176], digital accreditation of academic learning and achievements [177], and university fees payments.

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 complements IoT by providing data security, reliability and efficient management. Implementation of the public ledger would overcome the low storage capabilities of the IoT devices. Various works to develop secure lightweight IoT architecture based on the blockchain have been proposed in the literature [56,178,179]. Furthermore, blockchain can be used for smart appliances, smart contracts [180] and economic exchange between IoT devices and sensors.

6.2.5. Governance

The accountability, safety, and automation provided by the blockchain to handle public records could eventually obstruct the issues of fraud and corruption while making the government services more transparent and efficient. Blockchain aims at providing government services by developing applications such as digital personal identity of the citizens [181], birth, death and marriage certificates registration, smart-city development [182–184], virtual notary and proof of ownership [24],

e-residency [185], land management [186], decentralized voting systems [187–189], record tracking for the refugees, supply chain for the weapons of mass destruction, and disaster management [190].

7. Blockchain Issues and Possible Solutions

7.1. Scalability

The major issue that limits the growth of public blockchain is the scalability. The public network grows rapidly in terms of participants and data as there is no restriction for a user to join the network. The number of transactions and block validations increases with the increasing number of users. This leads to communication overhead which puts a burden on the network scalability.

Several works in the literature attempt to address the issue of scalability. [191] introduced the Segregated Witness (SegWit) to solve the issue of scalability in the Bitcoin network. This is by separating the transaction signature and the private key (generally the witness for a transaction) from the transaction data in a block by using two different chains. This increases the number of transactions in a block leading to an increase in throughput. [192] proposed the lighting network to solve the scalability issue. Lightning network is a layer on top of the blockchain network that allows users to perform transactions off-chain using channels without waiting for the block to be processed. However, both solutions to scalability were proposed for the Bitcoin network.

To improve the blockchain scalability beyond the Bitcoin network, [193] proposed a network based on the concept of sharding [194]. The nodes are divided into shards, and the transactions are distributed among the shards in parallel for verification. This increases the transactions throughput with increasing number of shards compared to a blockchain network. However, these proposals are still under development and are not compared in terms of performance with different existing blockchain networks.

7.2. High Energy Consumption

The energy consumed by the compute-intensive-based consensus protocols in the blockchain network is a major concern for the environment. Most of the blockchain network uses the compute-intensive protocol PoW as it is the most tested protocol ensuring a high level of security. A report by American magazine Grist stated that with the current mining power trend, the Bitcoin network by 2020 will consume the same amount of power that the entire world uses today. This energy consumption results in environmental hazard such as global warming and carbon footprints [195].

Various researchers proposed the use of an alternate consensus protocol and/or the use of energy-efficient hardware to tackle this issue. The compute-intensive-based protocols are replaced by the capability or voting-based protocols. Intel proposed and patented a new generation hardware accelerator that aims to reduce the mining power of the Bitcoin network by 35% [196].

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 verifications before it can be processed. This becomes time-consuming with a greater number of users, especially 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.

Developers and researchers have been working on increasing the performance of the blockchain technology when used at large-scale. New consensus mechanisms such as the byzantine fault tolerance [124], the delegated proof of stake [90], and the federated byzantine agreement [129] were proposed and used by different blockchain platforms. [197,198] proposed the use of linked transactions instead of linked blocks in the ledger to solve the problem of the transactions being delayed for

processing in a block until the block is complete, mined by the miners, verified by the validators and updated in the ledger. However, in the linked-transactions solution, the ledger is shared only among the peers participating in a transaction. This results in having different ledgers in the same network. Consequently, when the ledger of one peer is corrupted or the peer behaves maliciously, it is difficult to agree on the correct state of the ledger. This leads to a result similar to the issue of a single point of failure in a centralized peer-to-peer network.

To solve the issue of a single point of failure in a linked-transactions ledger, the blockchain development platform IOTA [199,200] proposed a ledger of transactions connected in a directed acyclic graph and maintained at each node. The directed graph of transactions is called a tangle in IOTA, with the vertices of the graph representing the transactions and the edges representing the validations (also known as approvals). A transaction which is not approved by any node is known as a tip. To get accepted in the network, a user must select two tips for the approval of the transaction submission [201]. These tips are known as a trunk transaction and a branch transaction. However, the tangled transactions are more prone to sybil attack, because it is possible to create random transactions by a malicious attacker 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 to the adoption of the technology. To address this issue, cloud providers such as IBM [202], and Amazon [203] are providing cloud-based blockchain templates to ease and automate the process of developing and deploying blockchain networks. Besides the cloud-based solution, there exist various offline blockchain development platforms (mentioned in this paper) which help in easy blockchain development and deployment.

7.5. Data Privacy

The public blockchain network has the property of pseudo-anonymity, which makes all the transactions data over the network visible to the public. While this feature helps in securing the network, it becomes a liability when used for sensitive data. Moreover, with multi-party transactions in the network, there are chances to track the real identity of a network participant [204].

Many technologies and methods such as the use of deterministic wallets [205], ring signature mixtures [206], ring confidential transactions [207], channels [208], and private network were proposed to achieve data privacy.

7.6. Lack of Governance

In the public blockchain network, where anyone can join the network, there is a lack of central governing authority that is needed to develop standard protocol and rules for transactions (especially financial transactions). The open source modification of the protocols makes it difficult to rely on them, especially when most of the network is dominated by malicious nodes. To solve this issue, organizations and companies are now moving towards the private blockchain networks, 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. These platforms have different architectures, programming languages, consensus protocols, and transactions flow. Consequently, applications built with different platforms cannot interoperate.

Standardization could help organizations to develop platforms to allow communications among heterogeneous blockchain networks. Seele is a blockchain platform enabling cross blockchain communications [209]. Openchain [59] and Elements [58] blockchain platforms provide this data transfer using a sidechain that interacts with the main blockchain. More work must be done to develop

specific standards for the development of platforms and consensus protocols to allow interoperability of blockchain applications.

7.8. Access to External Data

With the increasing adoption of the blockchain by real world applications (IoT for example), it becomes necessary for the blockchain network to communicate with external entities. Therefore, integrating services and relay network protocols [210] were introduced into the blockchain to link the network nodes to the outside world [211]. However, there are still some issues to address such as:

- How to ensure the reliability of the external data?
- How to reach consensus in case some of the nodes cannot reach the external data source due to network connectivity issues or if the data source becomes unavailable?
- Who will govern the third-party data sources?

8. Conclusions

The blockchain technology was introduced over a decade ago to perform peer-to-peer transactions of digital currencies between a group of untrusted network participants without the need for a third party. Over time, blockchain has evolved to develop decentralized applications beyond financial transactions in different fields. Consequently, various blockchain architectures and consensus protocols proliferated. Recently, the need for an open, flexible, scalable, and energy-efficient blockchain becomes crucial. This is due to the continuously evolving application requirements for better services in a large-scale collaborative ecosystem such as smart cities, social networking, governance, and smart healthcare with the ultimate goals for green computing and cost reduction. Consequently, blockchain architecture and consensus protocols have misaligned with the goals for a green collaborative digital ecosystem.

Initially implemented using a public network for an open participation with no access control to data, scalability, energy consumption, and security and privacy threats were major issues in the blockchain network. Later, blockchain architectures and consensus protocols were developed to enable the development of applications using a public network addressing the issues of security and privacy. However, one of the major challenges in those architectures and protocols is the high amount of energy consumption to ensure security. The energy consumption of the compute-intensive-based consensus protocols increases with increasing number of participants resulting in an adverse effect on the environment. Consequently, the capability-based or voting-based protocols were proposed as alternative solutions to reduce the energy consumption. However, these attempts to tackle the issue of high energy consumption led to less scalable and decentralized architectures and consensus protocols. A possible solution could be to develop a compute-intensive-based consensus protocol that is less computationally complex and more energy-efficient.

On the other hand, the inflexible and non-adaptive behavior of the current architectures and consensus protocols act as a barrier to serve the ultimate goals for a growing collaborative digital ecosystem. This is because current architectures and protocols target specific application domains without considering the future needs in a rapidly evolving collaborative ecosystem. Consequently, they fail behind flexible adaptation according to the application needs and require the modifications to 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 cannot be neglected while making the blockchain framework more energy-efficient and scalable. Currently, energy-efficient blockchain framework makes the network less decentralized and more prone to malicious attacks.

The objective of this paper is to further the research about blockchain in the context of smart cities collaborative ecosystem. In summary, it highlights the temporal evolution of the different blockchain architectures and consensus protocols, providing a retrospective analysis of their characteristics along

with their contributions and limitations. Further research is still needed to develop a blockchain framework that is open to large-scale 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.

Author Contributions: Conceptualization, L.I.; methodology, L.I.; investigation, L.I. and H.M.; writing-original draft preparation, L.I. and H.M.; writing-review and editing, L.I.; supervision, L.I.; project administration, L.I.

Funding: This work was funded by the Emirates Center for Energy and Environment Research of the United Arab Emirates University under Grant 31R101.

Acknowledgments: We thank the anonymous reviewers for their valuable comments which helped us improve the content, quality, and presentation of this paper.

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. *Int. J. Web Grid Serv.* **2018**, *14*, 352–375.
2. Nakamoto, S. Bitcoin: A Peer-To-Peer Electronic Cash System. 2008. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 22 September 2019).
3. Bitcoin Mining Consumes More Electricity a Year than Ireland | Technology | The Guardian. Available online: https://www.theguardian.com/technology/2017/nov/27/bitcoin-mining-consumes-electricity-ireland (accessed on 8 January 2019).
4. Stoll, C.; Klaaßen, L.; Gallersdörfer, U. The Carbon Footprint of Bitcoin. *Joule* **2019**, doi:10.1016/j.joule.2019.05.012.
5. Ahamad, S.; Nair, M.; Varghese, B. A survey on crypto currencies. In Proceedings of the 4th International Conference on Advances in Computer Science, AETACS, NCR, India, 13–14 December 2013; pp. 42–48.
6. Tschorsch, F.; Scheuermann, B. Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies. *IEEE Commun. Surv. Tutor.* **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. In Proceedings of the 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), Agadir, Morocco, 29 November–2 December 2016; IEEE: Piscataway, NJ, USA; 2016; pp. 1–6.
12. Karafiloski, E.; Mishev, A. Blockchain solutions for big data challenges: A literature review. In Proceedings of the IEEE EUROCON 2017—17th International Conference on Smart Technologies, Ohrid, Macedonia, 6–8 July 2017; IEEE: Piscataway, NJ, USA, 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 Gener. Comput. Syst.* **2017**, doi: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. *J. Netw. Comput. Appl.* **2019**, *126*, 45–58, doi: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. In Proceedings of the 2016 14th Annual Conference on Privacy, Security and Trust (PST), Auckland, New Zealand, 12–14 December 2016; pp. 745–752, doi:10.1109/PST.2016.7906988.
18. Nguyen, T.; Kim, K. A survey about consensus algorithms used in Blockchain. *J. Inf. Process. Syst.* **2018**, *14*, 101–128, doi:10.3745/JIPS.01.0024.
19. Wahab, A.; Mehmood, W. Survey of Consensus Protocols. *arXiv* **2018**, arXiv:1810.03357.
20. Remote Procedure Call—Wikipedia. Available online: https://en.wikipedia.org/wiki/Remote_procedure_call (accessed on 13 January 2019).
21. Web API—Wikipedia. Available online: https://en.wikipedia.org/wiki/Web_API (accessed on 13 January 2019).
22. Representational State Transfer—Wikipedia. Available online: https://en.wikipedia.org/wiki/Representational_state_transfer (accessed on 13 January 2019).
23. Merkle, R.C. A Digital Signature Based on a Conventional Encryption Function. In Proceedings of the Conference on the Theory and Applications of Cryptographic Techniques on Advances in Cryptology, CRYPTO '87, Santa Barbara, CA, USA, 23–27 August 1988; Springer: London, UK, 1988; pp. 369–378.
24. Swan, M. *Blockchain: Blueprint for a New Economy*; O'Reilly Media, Inc.: Newton, MA, USA, 2015.
25. Antonopoulos, A.M. *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*; O'Reilly Media, Inc.: Newton, MA, USA, 2014.
26. Secure Hash Algorithms. Available online: https://en.wikipedia.org/wiki/Secure_Hash_Algorithms (accessed on 7 January 2019).
27. Brewer, E. CAP twelve years later: How the" rules" have changed. *Computer* **2012**, *45*, 23–29.
28. 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* **2017**, arXiv:1710.09209.
29. Decker, C.; Wattenhofer, R. Information propagation in the bitcoin network. In Proceedings of the 2013 IEEE Thirteenth International Conference on Peer-to-Peer Computing (P2P), Trento, Italy, 9–11 September 2013; IEEE: Piscataway, NJ, USA; 2013; pp. 1–10.
30. Szabo, N. Smart contracts: Building blocks for digital markets. *Extropy: J. Transhumanist Thought* **1996**, *18*, 2.
31. Seele: Blockchain 4.0 or Marketing?—CoinAnnouncer. Available online: https://www.coinannouncer.com/seele-blockchain-4-0-or-marketing/ (accessed on 3 January 2019).
32. Pseudonymity. Available online: https://en.wikipedia.org/wiki/Pseudonymity (accessed on 18 January 2019).
33. Joshi, A.P.; Han, M.; Wang, Y. A survey on security and privacy issues of blockchain technology. *Math. Found. Comput.* **2018**, *1*, 121–147.
34. Buterin, V. *A Next-Generation Smart Contract and Decentralized Application Platform*—White Paper. 2014. Available online: http://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf (accessed on 22 September 2019).
35. Lai, R.; Chuen, D.L.K. Blockchain—From public to private. In *Handbook of Blockchain, Digital Finance, and Inclusion*; Elsevier, Amsterdam, The Netherlands, 2018; Volume 2, pp. 145–177.
36. Greenspan, G. MultiChain Private Blockchain—White Paper. 2015. Available online: Http://www.multichain.com/download/MultiChain-White-Paper.pdf (accessed on 22 September 2019).
37. Hyperledger Burrow. Available online: https://www.hyperledger.org/projects/hyperledger-burrow (accessed on 29 December 2018).
38. Chain Core. Available online: https://chain.com/docs/1.2/core/get-started/introduction (accessed on 29 December 2018).
39. Introduction—Sawtooth. Available online: https://sawtooth.hyperledger.org/docs/core/releases/1.0/introduction.html (accessed on 29 December 2018).
40. GitHub—HydraChain/hydrachain: Permissioned Distributed Ledger based on Ethereum. Available online: https://github.com/HydraChain/hydrachain (accessed on 3 January 2019).
41. Hyperledger Iroha—Hyperledger. Available online: https://www.hyperledger.org/projects/iroha (accessed on 29 December 2018).
42. The Burst Dymaxion. Available online: https://www.burst-coin.org/wp-content/uploads/2017/07/The-Burst-Dymaxion-1.00.pdf (accessed on 16 January 2019).

43. NEM White Paper. Available online: https://nem.io/wp-content/themes/nem/files/NEM_techRef.pdf (accessed on 16 January 2019).
44. 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. Available online: https://www.bigchaindb.com/whitepaper/ (accessed on 22 September 2019).
45. Quorum Overview. Available online: https://github.com/jpmorganchase/quorum/wiki/Quorum-Overview (accessed on 30 December 2018).
46. Durumeric, Z.; Kasten, J.; Bailey, M.; Halderman, J.A. Analysis of the HTTPS certificate ecosystem. In Proceedings of the 2013 conference on Internet Measurement Conference, Barcelona, Spain, 23–25 October 2013; ACM: New York, NY, USA, 2013; pp. 291–304.
47. Potlapally, N.R.; Ravi, S.; Raghunathan, A.; Jha, N.K. Analyzing the energy consumption of security protocols. In Proceedings of the 2003 international symposium on Low Power Electronics and Design, Seoul, Korea, 25–27 August 2003; ACM: New York, NY, USA, 2003; pp. 30–35.
48. Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; De Caro, A.; Enyeart, D.; Ferris, C.; Laventman, G.; Manevich, Y.; et al. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. In Proceedings of the Thirteenth EuroSys Conference, EuroSys '18, Porto, Portugal, 23–26 April 2018; ACM: New York, NY, USA, 2018; pp. 30:1–30:15, doi:10.1145/3190508.3190538.
49. Hyperledger Fabric—Hyperledger. Available online: https://www.hyperledger.org/projects/fabric (accessed on 29 December 2018).
50. Channels. Available online: https://hyperledger-fabric.readthedocs.io/en/release-1.3/channels.html (accessed on 29 December 2018).
51. Thakkar, P.; Nathan, S.; Vishwanathan, B. Performance Benchmarking and Optimizing Hyperledger Fabric Blockchain Platform. *arXiv* **2018**, arXiv:1805.11390.
52. Blockchain Platform | Oracle Cloud. Available online: https://cloud.oracle.com/en_US/blockchain (accessed on 14 January 2019).
53. Credits White Paper. Available online: https://credits.com/Content/Docs/TechnicalWhitePaperCREDITSEng.pdf (accessed on 29 December 2018).
54. LZ77. Available online: https://en.wikipedia.org/wiki/LZ77_and_LZ78 (accessed on 29 December 2018).
55. Huffman Coding. Available online: https://en.wikipedia.org/wiki/Huffman_coding (accessed on 29 December 2018).
56. Dorri, A.; Kanhere, S.S.; Jurdak, R. Towards an optimized blockchain for IoT. In Proceedings of the Second International Conference on Internet-of-Things Design and Implementation, Pittsburgh, PA, USA, 18–21 April 2017; ACM: New York, NY, USA, 2017; pp. 173–178.
57. Kousaridas, A.; Falangitis, S.; Magdalinos, P.; Alonistioti, N.; Dillinger, M. SYSTAS: Density-based algorithm for clusters discovery in wireless networks. In Proceedings of the 2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Hong Kong, China, 30 August–2 September 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 2126–2131.
58. Elements | elementsproject.org. Available online: https://elementsproject.org/ (accessed on 31 December 2018).
59. Overview of Openchain. Available online: https://docs.openchain.org/en/latest/general/overview.html (accessed on 29 December 2018).
60. Home | Lisk Documentation. Available online: https://lisk.io/documentation/home (accessed on 14 January 2019).
61. Quorum White Paper. Available online: https://github.com/jpmorganchase/quorum-docs/blob/master/Quorum%20Whitepaper%20v0.1.pdf (accessed on 30 December 2018).
62. Dwork, C.; Naor, M. Pricing via Processing or Combatting Junk Mail. In Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology, CRYPTO '92, Santa Barbara, CA, USA, 16–20 August 1992; Springer: London, UK, 1993; pp. 139–147.
63. Back, A. Hashcash—A Denial of Service Counter-Measure. 2002. Available online: http://www.hashcash.org/hashcash.pdf (accessed on 22 September 2016).
64. Brute-Force Search—Wikipedia. Available online: https://en.wikipedia.org/wiki/Brute-force_search (accessed on 18 January 2019).

65. Eastlake, D.; Jones, P. US Secure Hash Algorithm (SHA1); Technical Report, RFC Editor; 2001. Available online: http://www.faqs.org/rfcs/rfc3174.html (accessed on 22 September 2019).
66. Hashcash—Wikipedia. Available online: https://en.wikipedia.org/wiki/Hashcash (accessed on 18 December 2018).
67. U.S. Department of Commerce; National Institute of Standards and Technology. Secure Hash Standard. 2012. Available online: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf (accessed on 22 September 2019).
68. Bitcoin—Wikipedia. Available online: https://en.wikipedia.org/wiki/Bitcoin (accessed on 8 January 2019).
69. The Mystery Behind Block Time—FACILELOGIN. Available online: https://medium.facilelogin.com/the-mystery-behind-block-time-63351e35603a (accessed on 18 December 2018).
70. Gervais, A.; Karame, G.O.; Wüst, K.; Glykantzis, V.; Ritzdorf, H.; Capkun, S. On the security and performance of proof of work blockchains. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; ACM: New York, NY, USA, 2016; pp. 3–16.
71. Schrijvers, O.; Bonneau, J.; Boneh, D.; Roughgarden, T. Incentive compatibility of bitcoin mining pool reward functions. In Proceedings of the International Conference on Financial Cryptography and Data Security, Christ Church, Barbados, 22–26 February 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 477–498.
72. Kroll, J.A.; Davey, I.C.; Felten, E.W. The economics of Bitcoin mining, or Bitcoin in the presence of adversaries. In Proceedings of the Twelfth Workshop on the Economics of Information Security (WEIS 2013), Washington, DC, USA, 11–12 June 2013. Available online: https://www.econinfosec.org/archive/weis2013/papers/KrollDaveyFeltenWEIS2013.pdf (accessed on 22 September 2019).
73. Conti, M.; Kumar, E.S.; Lal, C.; Ruj, S. A Survey on Security and Privacy Issues of Bitcoin. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 3416–3452, doi:10.1109/COMST.2018.2842460.
74. Bitcoin Energy Consumption Index—Digiconomist. Available online: https://digiconomist.net/bitcoin-energy-consumption (accessed on 8 January 2019).
75. Litecoin—Open Source P2P Digital Currency. Available online: https://litecoin.org/ (accessed on 18 December 2018).
76. Dogecoin. Available online: https://dogecoin.com/ (accessed on 18 December 2018).
77. King, S. Primecoin: Cryptocurrency with prime number proof-of-work. *July 7th* **2013**, *1*, 6.
78. Ingham, A.E.; Ingham, A.E. *The Distribution of Prime Numbers*; Number 30; Cambridge University Press: Cambridge, UK, 1990.
79. Young, A.; Yung, M. Finding length-3 positive cunningham chains and their cryptographic significance. International Algorithmic Number Theory Symposium, Portland, OR, USA, 21–25 June 1998; Springer: Berlin/Heidelberg, Germany, 1998; pp. 289–298.
80. Forbes, T. Prime clusters and Cunningham chains. *Math. Comput. Am. Math. Soc.* **1999**, *68*, 1739–1747.
81. Ribenboim, P. *The New Book of Prime Number Records*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
82. Cladwell, C. Fermat Primality Test. Available online: https://primes.utm.edu/prove/prove2_1.html (accessed on 10 January 2019).
83. lifchitz, H. Generalization of Euler-Lagrange Theorem and New Primality Tests. Available online: http://www.primenumbers.net/Henri/us/NouvTh1us.htm (accessed on 10 January 2019).
84. Primecoin. Available online: http://primecoin.io/ (accessed on 10 January 2019).
85. Komodo White Paper. Available online: https://komodoplatform.com/wp-content/uploads/2018/05/2018-05-09-Komodo-White-Paper-Full.pdf (accessed on 25 January 2019).
86. Confirmation—Bitcoin Wiki. Available online: https://en.bitcoin.it/wiki/Confirmation (accessed on 24 January 2019).
87. Proof of Stake Instead of Proof of Work. Available online: https://bitcointalk.org/index.php?topic=27787.0 (accessed on 10 January 2019).
88. King, S.; Nadal, S. PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake; 2018. Available online: https://decred.org/research/king2012.pdf (accessed on 22 September 2019).
89. Whitepaper:Nxt—Nxt Wiki. Available online: http://nxtwiki.org/wiki/Whitepaper:Nxt (accessed on 10 January 2019).
90. Larimer, D. Delegated Proof-of-Stake Consensus. 2014. Available online: https://bitshares.org/technology/delegated-proof-of-stake-consensus (accessed on 23 January 2019).

91. bitshares.foundation/BitSharesBlockchain.pdf at master · bitshares-foundation/bitshares.foundation · GitHub. Available online: https://github.com/bitshares-foundation/bitshares.foundation/blob/master/download/articles/BitSharesBlockchain.pdf (accessed on 23 January 2019).
92. Nano—An Instant, Zero-Fee, Scalable Currency. Available online: https://nano.org/en (accessed on 23 January 2019).
93. Cardano—Home of the Ada Cryptocurrency and Technological Platform. Available online: https://www.cardano.org/en/home/ (accessed on 23 January 2019).
94. Ren, L. Proof of Stake Velocity: Building the Social Currency of the Digital Age. 2014 Available online: https://www.reddcoin.com/papers/PoSV.pdf (accessed on 22 September 2019).
95. Reddcoin Social Currency—Official Website. Available online: https://reddcoin.com/ (accessed on 10 January 2019).
96. Proof of Burn—Bitcoin Wiki. Available online: https://en.bitcoin.it/wiki/Proof_of_burn (accessed on 10 January 2019).
97. Slimcoin Whitepaper. Available online: http://www.doc.ic.ac.uk/~ids/realdotdot/crypto_papers_etc_worth_reading/proof_of_burn/slimcoin_whitepaper.pdf (accessed on 10 January 2019).
98. Slimcoin | A Cryptocurrency for the Long Term. Available online: http://slimco.in/ (accessed on 10 January 2019).
99. TGcoin. Available online: https://trade.tgco.in/ (accessed on 10 January 2019).
100. Dziembowski, S.; Faust, S.; Kolmogorov, V.; Pietrzak, K. Proofs of space. In Proceedings of the Annual Cryptology Conference, Santa Barbara, CA, USA, 16–20 August 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 585–605.
101. Bresson, E.; Canteaut, A.; Chevallier-Mames, B.; Clavier, C.; Fuhr, T.; Gouget, A.; Icart, T.; Misarsky, J.F.; Naya-Plasencia, M.; Paillier, P.; et al. Shabal, a Submission to NIST's Cryptographic Hash Algorithm Competition. 2008. Available online: https://www.cs.rit.edu/~ark/20090927/Round2Candidates/Shabal.pdf (accessed on 22 September 2019).
102. Space Coin. Available online: http://spacecoin.info/ (accessed on 10 January 2019).
103. Home—Chia Network. Available online: https://chia.net/ (accessed on 10 January 2019).
104. Burstcoin—The Linux of Blockchain. Available online: https://www.burst-coin.org/ (accessed on 10 January 2019).
105. Solana: A New Architecture for a High Performance Blockchain. Available online: https://solana.com/solana-whitepaper.pdf (accessed on 11 January 2019).
106. Proof of Importance. Available online: https://nem.io/xem/harvesting-and-poi/ (accessed on 18 January 2019).
107. Proof of Believability. Available online: https://github.com/iost-official/Documents/blob/master/Technical_White_Paper/EN/Tech_white_paper_EN.md (accessed on 18 January 2019).
108. IOST—UNLEASHING THE POWER OF BLOCKCHAIN. Available online: https://iost.io/ (accessed on 18 January 2019).
109. Proof-of-Authority—Wikipedia. Available online: https://en.wikipedia.org/wiki/Proof-of-authority (accessed on 23 January 2019).
110. Wiki/Proof-of-Authority-Chains.md at Master · paritytech/wiki · GitHub. Available online: https://github.com/paritytech/wiki/blob/master/Proof-of-Authority-Chains.md (accessed on 23 January 2019).
111. POA Network: Public Ethereum Sidechain with Proof of Autonomy Consensus by Independent Validators. Available online: https://poa.network/ (accessed on 23 January 2019).
112. Gochain-Whitepaper. Available online: https://gochain.io/gochain-whitepaper-v2.1.2.pdf (accessed on 23 January 2019).
113. Menlo One—Tools that Make Blockchain Work for Business. Available online: https://www.menlo.one/ (accessed on 23 January 2019).
114. The Second Coming of Blockchain | Intel® Software. Available online: https://software.intel.com/en-us/blogs/2017/02/14/the-second-coming-of-blockchain (accessed on 18 December 2018).
115. Sabt, M.; Achemlal, M.; Bouabdallah, A. Trusted Execution Environment: What It is, and What It is Not. In Proceedings of the 2015 IEEE Trustcom/BigDataSE/ISPA, Helsinki, Finland, 20–22 August 2015; Volume 1, pp. 57–64, doi:10.1109/Trustcom.2015.357.

116. Intel SGX Homepage | Intel® Software. Available online: https://software.intel.com/en-us/sgx (accessed on 18 December 2018).
117. Chen, L.; Xu, L.; Shah, N.; Gao, Z.; Lu, Y.; Shi, W. On Security Analysis of Proof-of-Elapsed-Time (PoET). Springer: Cham, Switzerland.
118. Chen, L.; Xu, L.; Shah, N.; Gao, Z.; Lu, Y.; Shi, W. On security analysis of proof-of-elapsed-time (poet). In Proceedings of the International Symposium on Stabilization, Safety, and Security of Distributed Systems, Boston, MA, USA, 5–8 November 2017; Springer: Piscataway, NJ, USA, 2017; pp. 282–297.
119. Hardin, G. The tragedy of the commons. *Science* **1968**, *162*, 1243–1248.
120. 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 Perform. Eval. Rev.* **2014**, *42*, 34–37.
121. Decred—Autonomous Digital Currency. Available online: https://www.decred.org/ (accessed on 11 January 2019).
122. Driscoll, K.; Hall, B.; Sivencrona, H.; Zumsteg, P. Byzantine fault tolerance, from theory to reality. In Proceedings of the International Conference on Computer Safety, Reliability, and Security, Edinburgh, UK, 23–26 September 2003; Springer: Piscataway, NJ, USA; 2003; pp. 235–248.
123. Lamport, L.; Shostak, R.; Pease, M. The Byzantine generals problem. *ACM Trans. Program. Lang. Syst. (TOPLAS)* **1982**, *4*, 382–401.
124. Castro, M.; Liskov, B. Practical Byzantine fault tolerance. In Proceedings of the 3rd Symposium on Operating System Design and Implementation (OSDI), New Orleans, Louisiana, USA, February 1999; Unisex Association: Berkeley, CA, USA, 1999; pp. 173–186.
125. Kwon, J. Tendermint: Consensus without mining. *Draft v. 0.6, fall* **2014**, *1*, 11.
126. Vukolić, M. The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication. International Workshop on Open Problems in Network Security, Zurich, Switzerland, 29 October 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 112–125.
127. Douceur, J.R. The sybil attack. In Proceedings of the International Workshop on Peer-to-Peer Systems, Cambridge, MA, USA, 7–8 March 2002; Springer: Berlin/Heidelberg, Germany, 2002; pp. 251–260.
128. NEO White Paper. Available online: https://docs.neo.org/en-us/whitepaper.html (accessed on 19 January 2019).
129. Schwartz, D.; Youngs, N.; Britto, A. The Ripple protocol consensus algorithm. *Ripple Labs Inc White Pap.* **2014**, *5*, 8.
130. Ripple—One Frictionless Experience To Send Money Globally | Ripple. Available online: https://ripple.com/ (accessed on 29 December 2018).
131. Mazieres, D. The Stellar Consensus Protocol: A Federated Model for Internet-Level Consensus. Stellar Development Foundation. 2015. Available online: https://www.stellar.org/papers/stellar-consensus-protocol.pdf (accessed on 29 December 2018).
132. Ongaro, D.; Ousterhout, J. In Search of an Understandable Consensus Algorithm. In Proceedings of the 2014 USENIX Annual Technical Conference (USENIX ATC 14), Philadelphia, PA, USA, 19–20 June 2014; USENIX Association: Berkeley, CA, USA, 2014; pp. 305–319.
133. Federated Consensus. Available online: https://chain.com/docs/1.2/protocol/papers/federated-consensus (accessed on 23 January 2019).
134. Tromp, J. Cuckoo Cycle: A memory-hard proof-of-work system. *IACR Cryptol. EPrint Arch.* **2014**, *2014*, 59.
135. Blockcerts: The Open Standard for Blockchain Credentials. Available online: https://www.blockcerts.org/ (accessed on 29 December 2018).
136. About—Blockcerts : The Open Standard for Blockchain Credentials. Available online: https://www.blockcerts.org/about.html (accessed on 29 December 2018).
137. Ismail, L.; Hameed, H.; AlShamsi, M.; AlHammadi, M.; AlDhanhani, N. Towards a Blockchain Deployment at UAE University: Performance Evaluation and Blockchain Taxonomy. In Proceedings of the 2019 International Conference on Blockchain Technology, Honolulu, HI, USA, 15–18 March 2019 ; ACM: New York, NY, USA, 2019; pp. 30–38.
138. MedRec. Available online: https://medrec.media.mit.edu/ (accessed on 29 December 2018).
139. SimplyVital Health. Available online: https://www.simplyvitalhealth.com/ (accessed on 29 December 2018).

140. Al Omar, A.; Rahman, M.S.; Basu, A.; Kiyomoto, S. Medibchain: A blockchain based privacy preserving platform for healthcare data. In Proceedings of the International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage, Guangzhou, China, 12–15 December 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 534–543.
141. Kuo, T.T.; Ohno-Machado, L. Modelchain: Decentralized privacy-preserving healthcare predictive modeling framework on private blockchain networks. *arXiv* **2018**, arXiv:1802.01746.
142. Roehrs, A.; da Costa, C.A.; da Rosa Righi, R. OmniPHR: A distributed architecture model to integrate personal health records. *J. Biomed. Inform.* **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. In Proceedings of the 2018 IEEE International Conference on Smart Computing (SMARTCOMP), Taormina, Italy, 18–20 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 49–56.
144. Mobile Bitcoin Wallet App—BTC & Cryptocurrency Wallet App | Abra. Available online: https://www.abra.com/ (accessed on 29 December 2018).
145. The Future of Energy | Blockchain, Transactive Grids, Microgrids, Energy Trading. Available online: https://lo3energy.com/ (accessed on 29 December 2018).
146. Every Product Has a Story | Provenance. Available online: https://www.provenance.org/ (accessed on 29 December 2018).
147. OpenBazaar. Available online: https://openbazaar.org/ (accessed on 29 December 2018).
148. About Us—Warranteer. Available online: http://www.warranteer.com/about-us (accessed on 29 December 2018).
149. Ujo Music. Available online: https://ujomusic.com/ (accessed on 29 December 2018).
150. Home | Yours. Available online: https://www.yours.org/ (accessed on 29 December 2018).
151. UBITQUITY—The First Enterprise Ready Blockchain-Secured Platform for Real Estate Recordkeeping | One Block At A Time. Available online: https://www.ubitquity.io/ (accessed on 29 December 2018).
152. Propy-Buy or Sell Investment Properties. Available online: https://propy.com/ (accessed on 29 December 2018).
153. Arcade City. Available online: https://arcade.city/ (accessed on 29 December 2018).
154. LaZooz. Available online: http://lazooz.org/ (accessed on 29 December 2018).
155. The Online Voting Platform of The Future—Follow My Vote. Available online: https://followmyvote.com/ (accessed on 29 December 2018).
156. Democracy Earth Foundation. Available online: https://www.democracy.earth/ (accessed on 29 December 2018).
157. uPort.me. Available online: https://www.uport.me/ (accessed on 29 December 2018).
158. Binded: Copyright Made Simple. Available online: https://binded.com/ (accessed on 29 December 2018).
159. Verisart. Available online: https://verisart.com/ (accessed on 29 December 2018).
160. Get Paperchain—Digital Media's First Global Marketplace for AR Financing. Available online: https://www.paperchain.io/ (accessed on 29 December 2018).
161. Po.et—The decentralized Protocol for Content Ownership, Discovery and Monetization of Media. Available online: https://www.po.et/ (accessed on 29 December 2018).
162. NASA Fund Researches the Potential of Blockchain Technology in Space. Available online: https://news.coinsquare.com/digital-currency/nasa-research-blockchain-in-space/ (accessed on 29 December 2018).
163. *Space Decentral: A Decentralized Autonomous Space Agency*; Technical Report. 2015. Available online: https://spacedecentral.net/White_Paper.pdf (accessed on 21 September 2019).
164. Coral. Available online: https://medium.com/spacedecentral/introducing-coral-an-open-lunar-space-program-702e293c9869 (accessed on 29 December 2018).
165. SpaceChain—Community-Based Space Platform. Available online: https://spacechain.com/ (accessed on 29 December 2018).
166. Qtum. Available online: https://qtum.org/en/ (accessed on 29 December 2018).
167. Smart Insurance Policy Powered by Blockchain. Available online: https://www.ibm.com/think/fintech/aig-ibm-standard-chartered-deliver-first-multinational-insurance-policy-powered-by-blockchain/ (accessed on 29 December 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. Available online: https://www-03.ibm.com/press/us/en/pressrelease/52970.wss (accessed on 19 December 2018).
169. Sharples, M.; Domingue, J. The blockchain and kudos: A distributed system for educational record, reputation and reward. In Proceedings of the European Conference on Technology Enhanced Learning, Lyon, France, 13–16 September 2016; Springer: Piscataway, NJ, USA, 2016; pp. 490–496.
170. Imagining a Blockchain University | Tom Vander Ark. Available online: https://www.gettingsmart.com/2018/06/imagining-a-blockchain-university/ (accessed on 19 December 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. Available online: https://chronobank.io/ (accessed on 19 December 2018).
173. Use for Blockchain in Libraries. Available online: https://ischoolblogs.sjsu.edu/blockchains/blockchains-applied/applications/ (accessed on 19 December 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. In Proceedings of the 2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL), Toronto, ON, Canada, 19–23 June 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. In Proceedings of the Ninth International Conference on Information and Communication Technologies and Development, Lahore, Pakistan, 16–19 November 2017; ACM: New York, NY, USA, 2017; p. 19.
177. Grech, A.; Camilleri, A.F. *Blockchain in Education*; Publications Office of the European Union: Luxembourg, 2017.
178. Dorri, A.; Kanhere, S.S.; Jurdak, R.; Gauravaram, P. Blockchain for IoT security and privacy: The case study of a smart home. In Proceedings of the 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kona, HI, USA, 13–17 March 2017; pp. 618–623, doi:10.1109/PERCOMW.2017.7917634.
179. Kravitz, D.W.; Cooper, J. Securing user identity and transactions symbiotically: IoT meets blockchain. In Proceedings of the Global Internet of Things Summit (GIoTS), Geneva, Switzerland, 6–9 June 2017; IEEE: Piscataway, NJ, USA 2017; pp. 1–6.
180. 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.
181. Rivera, R.; Robledo, J.G.; Larios, V.M.; Avalos, J.M. How digital identity on blockchain can contribute in a smart city environment. In Proceedings of the 2017 International Smart Cities Conference (ISC2), Wuxi, China, 14–17 September 2017; pp. 1–4, doi:10.1109/ISC2.2017.8090839.
182. Ibba, S.; Pinna, A.; Seu, M.; Pani, F.E. CitySense: Blockchain-oriented smart cities. In Proceedings of the XP2017 Scientific Workshops, Cologne, Germany, 22–26 May 2017; ACM: New York, NY, USA, 2017; p. 12.
183. Biswas, K.; Muthukkumarasamy, V. Securing Smart Cities Using Blockchain Technology. In Proceedings of the 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), Sydney, NSW, Australia, 12–14 December 2016; pp. 1392–1393, doi:10.1109/HPCC-SmartCity-DSS.2016.0198.
184. 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.
185. Sullivan, C.; Burger, E. E-residency and blockchain. *Comput. Law Secur. Rev.* **2017**, *33*, 470–481, doi:10.1016/j.clsr.2017.03.016.
186. Pichel, F. Blockchain for land administration. *GIM Int.* **2016**, *30*, 38–39.
187. Meter, C. Design of Distributed Voting Systems. *arXiv* **2017**, arXiv:1702.02566.
188. Noizat, P. Chapter 22—Blockchain Electronic Vote. In *Handbook of Digital Currency*; Chuen, D.L.K., Ed.; Academic Press: San Diego, CA, USA, 2015; pp. 453–461, doi:10.1016/B978-0-12-802117-0.00022-9.

189. Hjalmarsson, F.P.; Hreioarsson, G.K.; Hamdaqa, M.; Hjalmtysson, G. Blockchain-Based E-Voting System. In Proceedings of the 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), San Francisco, CA, USA, 2–7 July 2018; Volume 00; pp. 983–986, doi:10.1109/CLOUD.2018.00151.
190. Panesir, M.S. Blockchain Applications for Disaster Management and National Security. Ph.D. Thesis, State University of New York at Buffalo, Buffalo, NY, USA, 2018.
191. SegWit. Available online: https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki (accessed on 31 December 2018).
192. Poon, J.; Dryja, T. The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments Technical Report, Technical Report (Draft). 2015. Available online: https://lightning.network (accessed on 21 September 2019).
193. Zilliqa. Available online: https://docs.zilliqa.com/positionpaper.pdf (accessed on 31 December 2018).
194. Sharding. Available online: https://www.investopedia.com/terms/s/sharding.asp (accessed on 31 December 2018).
195. Bitcoin Could Cost Us Our Clean-Energy Future | Grist. Available online: https://grist.org/article/bitcoin-could-cost-us-our-clean-energy-future/ (accessed on 31 December 2018).
196. BITCOIN MINING HARDWARE ACCELERATOR WITH OPTIMIZED MESSAGE DIGEST AND MESSAGE SCHEDULER DATAPATH. Available online: 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 31 December 2018).
197. Brown, R.G.; Carlyle, J.; Grigg, I.; Hearn, M. Corda: An introduction. *R3 CEV* **2016**, *1*, 15.
198. Corda. Available online: https://docs.corda.net/ (accessed on 30 December 2018).
199. Popov, S. The tangle. *cit. on* **2016**, 131. Available online: http://tanglereport.com/wp-content/uploads/2018/01/IOTA_Whitepaper.pdf (accessed on 29 December 2018).
200. What Is IOTA. Available online: https://docs.iota.org/introduction/what-is-iota (accessed on 29 December 2018).
201. Tip Selection—IOTA Docs. Available online: https://docs.iota.org/introduction/tangle/tip-selection (accessed on 29 December 2018).
202. Blockchain on Bluemix: IBM Blockchain Blog. Available online: https://www.ibm.com/blogs/blockchain/category/blockchain-on-cloud/ (accessed on 25 January 2019).
203. Blockchain on AWS. Available online: https://aws.amazon.com/blockchain/ (accessed on 25 January 2019).
204. Genkin, D.; Papadopoulos, D.; Papamanthou, C. Privacy in Decentralized Cryptocurrencies. *Commun. ACM* **2018**, *61*, 78–88, doi:10.1145/3132696.
205. Deterministic Wallet—Bitcoin Wiki. Available online: https://en.bitcoin.it/wiki/Deterministic_wallet. (accessed on 2 January 2019).
206. Ring Signature—Wikipedia. Available online: https://en.wikipedia.org/wiki/Ring_signature (accessed on 2 January 2019).
207. Noether, S.; Mackenzie, A.; The Monero Research Lab. Ring confidential transactions. *Ledger* **2016**, *1*, 1–18.
208. Coleman, J.; Horne, L.; Xuanji, L. Counterfactual: Generalized State Channels; 2018. Available online: https://l4.ventures/papers/statechannels.pdf (accessed on 18 September 2019).
209. Seele. Available online: https://seele.pro/ (accessed on 3 January 2019).
210. Hirn, M. Rlay: A Decentralized Information Network. Available online: https://rlay-project.github.io/rlay.com/rlay-whitepaper.pdf (accessed on 2 January 2019).
211. Corda Nodes. Available online: https://docs.corda.net/corda-test-networks.html (accessed on 30 December 2018).

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).