

# What is a Blockchain? A Definition to Clarify the Role of the Blockchain in the Internet of Things

Lorenzo Ghiro,<sup>1</sup> Francesco Restuccia,<sup>2</sup> Salvatore D’Oro,<sup>2</sup> Stefano Basagni,<sup>2</sup>  
Tommaso Melodia,<sup>2</sup> Leonardo Maccari,<sup>3</sup> Renato Lo Cigno<sup>4</sup>  
<sup>1</sup>University of Trento, Italy, <sup>2</sup>Northeastern University, USA,  
<sup>3</sup>University of Venice, Italy, <sup>4</sup>University of Brescia, Italy

## Abstract

The use of the term *blockchain* is documented for disparate projects, from cryptocurrencies to applications for the Internet of Things (IoT), and many more. The concept of blockchain appears therefore blurred, as it is hard to believe that the same technology can empower applications that have extremely different requirements and exhibit dissimilar performance and security. This position paper elaborates on the theory of distributed systems to advance a clear definition of blockchain that allows us to clarify its role in the IoT. This definition inextricably binds together three elements that, as a whole, provide the blockchain with those unique features that distinguish it from other distributed ledger technologies: *immutability*, *transparency* and *anonymity*. We note however that immutability comes at the expense of remarkable resource consumption, transparency demands no confidentiality and anonymity prevents user identification and registration. This is in stark contrast to the requirements of most IoT applications that are made up of resource constrained devices, whose data need to be kept confidential and users to be clearly known. Building on the proposed definition, we derive new guidelines for selecting the proper distributed ledger technology depending on application requirements and trust models, identifying common pitfalls leading to improper applications of the blockchain. We finally indicate a feasible role of the blockchain for the IoT: myriads of local, IoT transactions can be aggregated off-chain and then be successfully recorded on an external blockchain as a means of public accountability when required.

## 1 Introduction

The *blockchain* came into the limelight with the advent of the Bitcoin cryptocurrency, by far the most successful blockchain application, which in January of 2021 set its new record of market capitalization exceeding 758 billions of US dollars. The blockchain features observed in Bitcoin, i.e., decentralization, resistance to powerful cyberattacks and preservation of users privacy, raised the enthusiasm of many research communities. This enthusiasm led to an extremely large number of disparate proposals for using the blockchain in many different applications, including Supply Chain Management [1–4], E-Voting [5–12], Smart Grid [13–20], Healthcare [21–24], Banking [25, 26], Smart Cities [27–29], and even Vehicular and Aerial Networks [30–48]. Surveys abound on the efforts of applying the blockchain also to the many expressions of the Internet of Things (IoT) [49–57]. This vast application range makes

the blockchain seem a *universal* technology, no longer limited only to cryptocurrencies but also capable to empower most IoT applications, addressing their multiple vulnerabilities [58].

This apparent universality of the blockchain looks suspicious, suggesting that the term is used with many different meanings. Considering that the Bitcoin blockchain currently supports the validation of less than 10 Transactions per Second (TPS) and exhibits a power consumption similar to that of an industrialized country such as Ireland [59], it is dubious that it can support the millions of IoT TPS [60] and meet the typical IoT power constraints. In fact, we notice that moving to application domains different from cryptocurrencies, the original characteristics of the Bitcoin blockchain have been diluted, if not completely transformed, leading to possible misunderstanding and confusion. On the one hand, we have the *permissionless* blockchains like Bitcoin [61], celebrated for their Proof of Work (PoW)-based cryptographical security, their decentralization and strict privacy defense through anonymity. On the other hand, many proposed “blockchains” are *permissioned*, requiring user identities to be registered with some trusted authority (or consortium) and whose internal security does not depend on some hard cryptographical problem like the PoW. As a consequence, the term blockchain appears to be overloaded, and therefore ambiguous, as it is used to indicate ledger technologies that under the hood obtains security, performance and decentralization in completely different ways.

This position paper analyzes the multiple technologies professed under the term *blockchain* and proposes a clear definition of blockchain that allows us to argue about its role in the IoT. Building on the theory of distributed systems and on the critical analysis of current blockchain applications, the definition identifies three elements that, only when combined together, give to blockchains their specific features of openness, decentralization, security and ability to preserve user privacy. These three elements are: relying on a STRONG DISTRIBUTED CONSENSUS PROTOCOL, which makes the blockchain immutable, hence secure from tampering attacks, and further frees the system from centralized trusted authorities (e.g., banks); maintaining a FULL & PUBLIC HISTORY OF TRANSACTIONS, which permits their distributed and completely transparent validation, and being OPEN TO ANONYMOUS USERS, thus allowing blockchains to preserve users privacy.

A definition for blockchain based on these three elements results quite restrictive, amplifying the voice of those who also advocate a strict definition of blockchain [62–64]. However, clarifying the nature of the blockchain with this clear definition allows us to argue that the blockchain is not the universal and limitless technology that may seem to be. In fact, we

state that blockchains are beneficial only for a limited range of applications, and that their integration into the IoT domain is not appropriate. We substantiate these arguments drawing new guidelines for the proper adoption of the blockchain, suggesting clearly when the blockchain should *not* be adopted. We do so highlighting a list of common scenarios where the applications requirements conflict with the blockchain characteristics. For example, the use of a blockchain for storing sensitive information is a pitfall, because the blockchain immutability would prevent the compliance with regulations that demand user data to be erasable upon request.

The remainder of this paper includes a review of the fundamental principles of the blockchain (Section 2) and of its theoretical roots, namely, distributed consensus protocols (Section 3). We acknowledge that the analysis proposed in this paper is partial, meaning that the goal of the paper is to *restrict* the meaning of blockchain, rather than building an exhaustive list of all its possible meanings and uses, as a standard survey would do. Still, we provide abundant academic references in support of our definition and highlight that there are only benefits—for the scientific community, industry and practitioners at large—to restrict and disambiguate the use of this term and rather using different ones, inventing them if necessary, to identify technologies and solutions that are far away from the original use of the term. In Section 4 we condense the specific features of blockchains into a connotative definition, based on which we build the guidelines for the proper adoption of the blockchain technology. We then discuss popular applications that contrast with the proposed definition of blockchain in Section 5, explaining which application requirements clash with the blockchain features. In Section 6 we indicate a possible role for blockchains in the IoT, namely, they can play as complementary (external) ledger services. Final remarks are drawn in Section 7.

## 2 Blockchain Fundamentals

Figure 1 is proposed to gently introduce the blockchain fundamentals starting from an illustration of the life-cycle of a transaction, which will be finally registered in a blockchain. Everything starts when a transaction is issued, e.g., because a smart device is querying a remote service and pays to access the data. The transaction is announced in the P2P network and received by validator nodes. These nodes run a consensus protocol to decide about the validity of the transaction. If they reach a consensus on the fact that the device really owns the resources that is about to spend, then the transaction is considered valid. If so, it is grouped with others recently approved, forming a new block of transactions that will be registered in the ledger by appending it to the blockchain. At the end, the success of the transaction is notified to the users and the data is transferred to the device.

At first glance, the blockchain can be defined as the plain data structure used to record transactions. However, from a broader perspective, a blockchain can be considered a distributed system that in general includes:

- A *Peer-to-Peer (P2P) network* made of all those nodes that either read or cooperatively write transactions in the blockchain, and

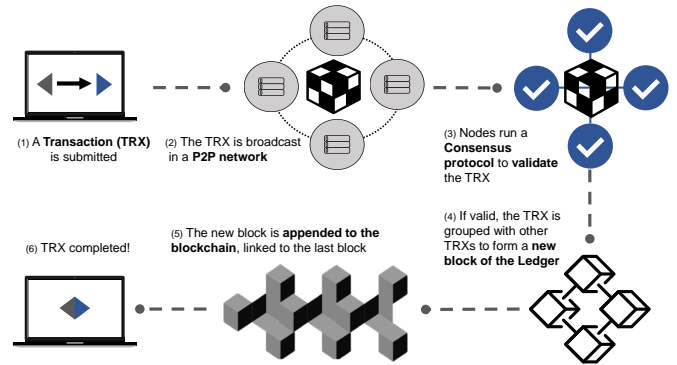


Figure 1: Processing of a transaction before storage in the blockchain.

- a *consensus protocol*, namely, a set of policies agreed upon and implemented by all nodes, which are the rules that regulate which and how new transactions can be added to the blockchain.

A blockchain turns out to be a possible implementation of a Shared Ledger. The group of entities allowed to write new transactions in the Shared Ledger, appending them to the blockchain, can vary from few selected and authenticated users up to any anonymous user. These different writing privileges depend on the rules of the chosen consensus protocol, which are decisive to determine if the resulting Shared Ledger will be *public* or *private*.

**Public or Permissionless Ledger** In a public (permissionless) ledger, the record of transactions is public and the consensus protocol is open to anybody. This means that i) anyone in the world can verify the correctness of the ledger and ii) even anonymous strangers without explicit permission can join the network and participate in the validation process of transactions, provided only that they comply with the consensus protocol. The absence of any form of control on users, that are not accountable as they are anonymous, is an issue for the security of the ledger. To counter this lack of trust and still ensure security, the usual consensus protocol of a permissionless ledger imposes stringent conditions to be met upon proposing a new block of transactions. Such conditions are so severe that, somehow, prove the honest commitment of the proposer. For example, in both Bitcoin and Ethereum—the most iconic permissionless blockchains—the proposer of a new block of transactions must provide the so-called Proof of Work (PoW), which is the solution to a very hard cryptographic problem. This mechanism secures the ledger discouraging malicious users, but hampers the ledger performance as well. Consider, for example, that the number of TPS processed by Bitcoin and Ethereum is on average below 20 TPS, a very limited throughput if compared to the tens of thousands processed by Visa [65]. Moreover, in the Visa platform transactions are recorded sequentially, not in blocks. For this reason the transaction latency, i.e., the interval of time between submission and recording of a transaction, can be kept down to at most a few seconds with Visa, while in Bitcoin the same latency averages tens of minutes and can grow up to several hours.

**Private or Permissioned Ledger** Private ledgers arose as an attempt to improve performance and to have more control

on users. These ledgers are typically implemented by big corporations or banks, so to have a common platform to share business information among few and well known partners. A shared and mutual level of trust can be given for granted, as only registered (hence accountable) entities have the permission to write data into the blockchain. The security of permissioned blockchains depends therefore on classical authentication mechanisms rather than on the mathematical strength of techniques such as the PoW. The trust model resulting from permissions allows blockchain managers to replace the resource-hungry consensus protocols of permissionless blockchains with more traditional, efficient, and faster ones. Such faster consensus protocols are necessary to support critical business operations, whose recording cannot tolerate the typical low transaction rates and high inefficiencies of permissionless ledgers.

For the first time, permissionless ledgers free users from trusted authorities, such as a Public Key Infrastructure (PKI) or banks, with the added novelty that the validation process happens transparently in public, without uncovering the identity of users. The trust required to maintain a public, permissionless ledger open to anonymous users is given by the consensus protocol only. A permissionless blockchain can thus be considered as a *trust builder in a trustless network* and the enabler of an *open, privacy-preserving, disintermediated marketplace*.

## 2.1 The Need of the Transactions History

Validators need the history of transactions to determine who and how many resources each user owns, an indispensable knowledge to validate new transactions. However, building this history in a distributed system is complicated by the double spending problem, briefly described below.

### Double Spending Problem

Two transactions that spend the same resources may be processed in different order by distinct validators spread across a P2P network. This is because of different propagation delays in the network (Figure 2). At this point, it is crucial for validators to find an agreement on the order of transactions to determine which of the two came in first, and should be considered valid, and which came in second and should be rejected.

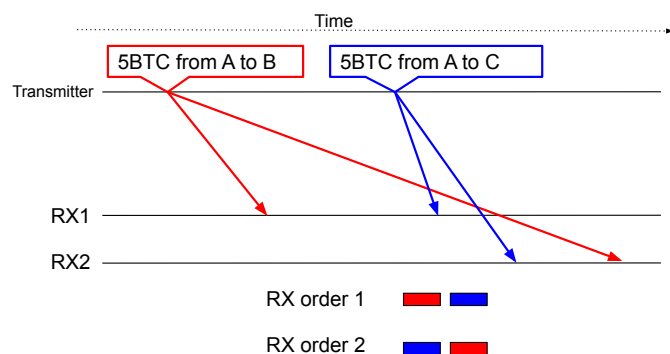


Figure 2: Example of how propagation delays may lead to two different orders of reception at distinct validators. Validators need to run a consensus protocol to find an agreement on the order of transactions. In this example if A owns only 5BTC then one of the two transactions must be rejected because it would represent a double spending.

This fundamental problem is also known as *Distributed Consensus Problem*. An equivalent problem is the implementation of a distributed timestamp server able to sort transactions in an

indisputable chronological order. The blockchain has been introduced in 2008 by the mysterious author of Bitcoin, Satoshi Nakamoto, exactly as a way to implement a distributed timestamp server that assigns timestamps to (blocks of) transactions, this way establishing their history.

However, the history of transactions may not be enough for a correct validation. Indeed, a malicious user can alter the content of a block to repudiate an unwanted transaction, ultimately falsifying the validation procedure. To fend off falsification attacks a blockchain must be:

- *Tamper-proof*: i.e., made so that is easy to verify that the registered transactions have not been manipulated after their recording, and it should be likewise easy to determine if these have been actually altered in a second instance of time.
- *Immutable*: a blockchain-based ledger should adequately word off tampering attacks.

The tamper-proof property of blockchains is achieved by a clever embedding of Cryptographic Hash Functions (CHF) into the blockchain data structure, as explained in Section 2.2.

## 2.2 The Blockchain Data Structure

CHFs are crucial to make blockchains tamper-proof. To become tamper-proof, transactions must be grouped into blocks including few other information as shown in Figure 3.

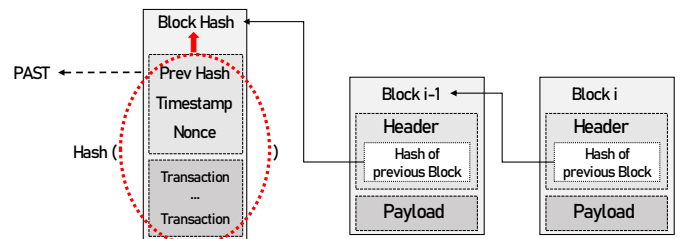


Figure 3: The structure of a blockchain.

For example, in Bitcoin a block is valid if, hashing its content, the produced fixed-length digest<sup>1</sup> exhibits a predefined number of leading zeros. This digest is said to be the “Block Hash.” The Block Hash must be a number lower than a given target, a target that can be changed to adjust the difficulty [66] of finding a valid Block Hash. Finding a valid Block Hash can become an hard problem considering the random nature of CHFs and the nodes strategy for generating new blocks, which is the following.

At first, a validator groups together some recent transactions and assigns them a timestamp. Then, to further enforce the time dependency between blocks, it includes in the new block the Block Hash of the last block he is aware of. The references to previous blocks constitute the *chaining* of blocks. Finally, a validator guesses a random value (the *nonce*), includes it in the new block, and applies the hash function to all these information to compute a new digest, which becomes a candidate Block Hash. This digest can be smaller than the target, thus not valid. A node is usually forced to retry with as many different random nonces as possible, until it produces (via brute force) a valid Block Hash.

<sup>1</sup> A message digest is a fixed size numeric representation of the contents of a message, as computed by a hash function.

If someone tries to tamper block  $i$ , the attack will invalidate its Block Hash with high probability. Because of blocks chaining also block  $i + 1$  gets invalidated and, with a domino effect, all blocks following block  $i$  get invalidated as well. This mechanism makes the blockchain a *tamper-proof* technology.

### 2.3 Proof of Work (PoW)

Consider that the CHF mandated by the Bitcoin protocol is double-SHA256, which produces digests of 256 bits and, at the current Bitcoin difficulty level, the first 77 bits must be zero. The probability for a random nonce to be valid can be approximately computed as a function of the required number of leading zeros, that we call  $Z$ . The set of all possible digests that the double-SHA256 function can generate has a cardinality of  $2^{256}$ . Only digests that have  $Z$  leading zeros are valid, so the cardinality of the set of valid digests is given by  $2^{256-Z}$ . The probability  $P(n)$  for a random nonce to produce a valid digest is therefore:

$$P(n) = \frac{2^{256-Z}}{2^{256}} = \frac{1}{2^Z}. \quad (1)$$

For  $Z = 77$ , then  $P(n) \approx 6.62 \times 10^{-24}$ .

When a node finds a valid nonce, it can show it to all other nodes in the P2P network as a *proof of work* (PoW), i.e., as a proof of the effort (computing power and ultimately energy in this case) that this node has spent to find such nonce. By showing a valid nonce, the node can claim the reward that the Bitcoin protocol assigns to nonce discoverers, which is the right of including a transaction that generates new Bitcoins and transfers them to the discoverer digital wallet. These rewards incentivize nodes in the hard task of discovering the very rare valid nonces. Those nodes that are constantly at work looking for valid nonces are metaphorically called “miners.” Asking miners to produce a PoW for building a valid block is an important mechanism to control the generation interval of new blocks (Section 2.4) and to secure the blockchain (Section 2.5).

### 2.4 Block Generation Interval

The difficulty of the PoW is tuned so that, in the whole P2P network of miners, a valid block is produced on average every 10 minutes. To keep a constant Block Generation Interval ( $B_{GI}$ ), the difficulty of the PoW must be tuned according to the miners computing power, which has remarkably increased during the years (Figure 4), reaching the record of 166 658 millions of Tera-hash computed per second (TH/s).

The  $B_{GI}$  must be kept high to avoid the simultaneous production of two blocks as far as possible. Two blocks are considered “simultaneous” if the second one is generated within the average Block Propagation time ( $B_P$ ), which for any P2P overlay based on transactions is in the orders of seconds to maximum tens of seconds [67]. Simultaneous blocks are problematic because their almost contemporaneous proposal divides the nodes of the Bitcoin network in two parties that will append the two different blocks to the blockchain, forming two branches. This situation can be represented by a bifurcation of the blockchain and is called a “fork.” When a fork occurs, it means that there is no distributed consensus on block order anymore, thus there is no agreement on the order of transactions. Without this agreement, the system is exposed again to Double Spending attacks.

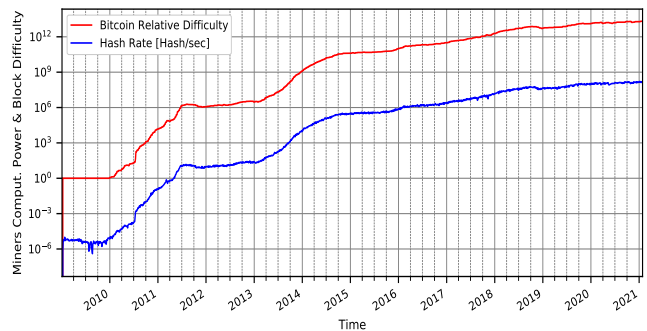


Figure 4: Evolution of the Bitcoin network computing power, measured in hash per second (shown in logarithmic scale). Over time the block difficulty has been adjusted to keep a constant average block production rate. Statistics are taken from [blockchain.com](https://blockchain.com)

Usually forks are transient and are cleared as soon as another block is presented, making one branch longer than the other. This mechanism is called “*the longest-chain rule*” [68, 69], and it is inherent to the blockchain technology, thus representing a constraint in the design of blockchain based systems. The rule also implies that orphan blocks that do not end up to be part of the longest chain are not valid: these blocks are also called *stale blocks*. The transactions included in the stale blocks, reward as well, will not be considered valid. Miners do not want to waste resources working on blocks that will not be part of the longest chain, therefore, they immediately switch to a longer branch as soon as they notice such one, increasing this way their chances of winning a more secure reward [68]. This ensures that the majority of miners always work on the longest branch.

### 2.5 The Security of the PoW

While block hashing and chaining make blockchains tamper-proof, as explained in Section 2.2, the PoW is key to make blockchains *immutable*. Consider a malicious user that performed a transaction spending a considerable amount of cryptocurrency. This transaction was included in a given block, say, block  $i$ , and the user now wants to revoke it. One way to revoke this transaction is to cancel it from block  $i$ , but this way block  $i$  would get invalidated and, by hash chaining, also all the following blocks: the tamper-proof property of blockchains defuses this kind of attack. Another strategy exploits the longest-chain rule and consists in generating a second and longer chain of blocks that, starting from block  $i - 1$ , would replace the current chain that contains block  $i$ .

This attack that exploits the longest-chain rule is clearly a daunting task. To be successful, the malicious user must beat all the other miners that during the attack keep using their collective computing resources to extend the blockchain with new blocks. In general, the average probability  $p_a$  of being the first to create and add a new block is the fraction of the computing resources controlled by a user, and block mining can be considered independent, so that the probability of adding  $N$  consecutive blocks is  $p_a^N$ , as shown in Figure 5 for different computing power ratios.<sup>2</sup>

<sup>2</sup> For each new block, all miners start competing to find a valid nonce almost at the same time, i.e., when the hash name of the last block is revealed and broadcast in the peer-to-peer network. For this reason, all the “races for the next block” can be considered independent.

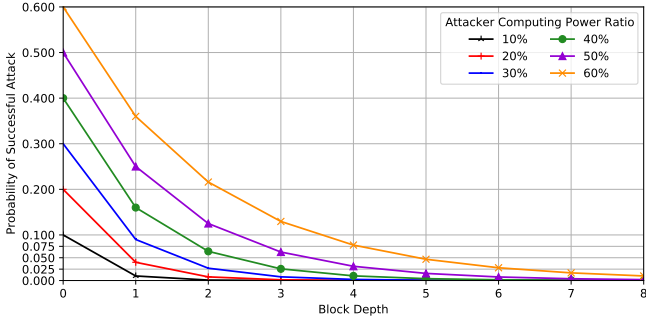


Figure 5: Probability for an attacker that owns different amounts of computing resources to tamper a block, as function of the block depth. The depth of a block  $b$  is the number of blocks that have been added to the blockchain after  $b$ . The success probability decays exponentially fast, vanishing for blocks that are deep in the blockchain.

Some believe that owning the majority of the computing power ensures the control of the network through the so called “50+1% attack.” The simple analysis above shows clearly that this is not enough, in general, to tamper any arbitrary block included in the history of transactions. Rather, the existence of an extremely powerful user may lead to a biased system where this user decides with high probability which transactions should be recorded and which others should be ignored from now on in the future, but the immutability of the deepest blocks of the blockchain is not harmed. In fact, Bitcoin uses block depth as *confirmation*, and recommends users to consider a transaction to be final only if it has been confirmed by 6 subsequent blocks [70], a number that guarantees it is almost unassailable.

## 2.6 Mining Overhead and Stale Rate

The *Average Mining Overhead (AMO)* denotes the percentage of the network computing power that, on average, does not contribute to the growth of the main chain, but rather leads to the generation of stale blocks.

*Def. 2.1: Average Mining Overhead*

$$AMO(R, B_P, B_{GI}) := (1 - R) \times \frac{B_P}{B_{GI}}$$

- $R \in (0, 1]$  is the ratio of the collective network computing power controlled on average by a successful miner. The complementary computing power ratio controlled by unsuccessful miners is  $1 - R$ ;
- $B_P$  is the average block propagation time;
- $B_{GI}$  is the average block generation time interval.

Definition 2.1 states that, in each block generation interval  $B_{GI}$ , unsuccessful miners waste their computing power  $(1 - R)$  for  $B_P$  time, i.e., until they learn about the newly proposed block. By definition, the AMO contributes to the generation of stale blocks. Hence, it is proportional to the *stale rate*, trivially defined by the ratio between the number of stale blocks and the total number of generated blocks.

*Def. 2.2: Stale Rate*

$$\frac{\# \text{Stale Blocks}}{\# \text{Stale} + \text{Final Blocks}} \propto AMO(R, B_P, B_{GI})$$

The stale rate (Definition 2.2) is considered an indicator of the security level of a PoW-based blockchain, because a higher

stale rate means that a blockchain is more exposed to chain replacement and eclipse attacks [71]. Notice also that, fixing the other parameters ( $R$  and  $B_P$ ), increasing the  $B_{GI}$  leads to a lower stale rate, enhancing the blockchain security. The high  $B_{GI}$  (in the order of several minutes) chosen by many blockchains for cryptocurrencies [72] represents the effort of enhancing the mining efficiency and security by trading transaction throughput and confirmation time. Notice also that a higher  $B_{GI}$  enhances the miners profits, because reducing the mining overhead implies that investments on mining equipment become in percentage more profitable.

## 2.7 Power consumption of the PoW

Section 2.6 highlighted how one can increase the security of a blockchain extending the Block Generation Interval ( $B_{GI}$ ). An extension of the  $B_{GI}$  can be immediately achieved increasing the mining difficulty: this way the cryptographical puzzle necessary to produce a valid block becomes harder, thus more time-consuming but also power-hungry. The effort of setting up an exceptionally secure blockchain resulted, in Bitcoin, in a PoW that has become extraordinarily power-hungry [59, 73]. The power consumption of the Bitcoin network in 2018 were estimated to be 2.55 GW, and forecast to reach 7.67 GW in the future. This requirement is comparable to the energy demand of a whole country such as Ireland [59], so it is hard to think that a blockchain secured by the PoW could be integrated in the constrained domain of the IoT.

## 3 Distributed Consensus Protocols

Section 2 posed the distributed consensus problem on the order of transactions and explained how the PoW solves it. The PoW advantages are many: it is extraordinarily secure, fully distributed, and user-agnostic. In fact, users can participate to a PoW-based consensus without registering their identity with some trusted registrar or bank, but just providing some computing power. Ultimately the PoW i) protects the user privacy and ii) free users from trusted authorities. The popularity of blockchains, above all with cryptocurrencies, is most probably grounded in these two key aspects. However, the PoW imposes also serious limitations in terms of transaction latency, throughput and power consumption.

It can be observed that consensus protocols are a crucial component for a Shared Ledger: performance, consistency, policies of governance, security, and tolerance to failures are all properties of a Shared Ledger that depend on the selected consensus protocol, rather than on the data structure used to record transactions. A question arises: Is it possible to design a consensus protocol that preserves the PoW advantages and, at the same time, avoids its drawbacks so to meet the typical requirements of IoT applications?

The rest of this section revises the general limits of distributed systems, which constitute theoretical bounds for the design of consensus protocols in general, and especially for IoT applications. We first summarize these fundamental theorems valid for any distributed system (Section 3.1), then we explore the trade-offs inherent to the blockchain technology (Section 3.2). Finally, in Section 3.3, we briefly review consensus protocols in search of alternatives to the PoW.

### 3.1 Limiting Theorems for Consensus

It is well known that it is impossible to achieve consensus in distributed systems in the presence of faulty nodes and unreliable communication channels [74]. The proof is based on this intuition: every time a consensus is close to be achieved among distributed agents, then a node or a communication failure may occur preventing the termination of consensus forever. This impossibility proof is in tight relation with another fundamental pillar of distributed systems, i.e., the *Consistency, Availability and Partition tolerance* (CAP) theorem [75]. The CAP theorem states that whenever a system gets *Partitioned*, then only two options are available: i) grant *Consistency* by safely blocking the system to fix the failures; or ii) keep processing transactions favoring *Availability*, with the risk that the two conflicting transactions (e.g., Double Spending ones) could be recorded, one per partition. This theorem is usually illustrated with a triangle with vertexes occupied by the three properties, stating that only two out of the three properties can be satisfied at the same time.

Both the impossibility proof and the CAP theorem may be considered only mildly relevant, as they are valid only for ill-behaving systems, while in practice a system is built to work properly for most of its lifetime. However, they are the anteroom for the definition of two (almost equivalent) tradeoffs of tremendous practical importance.

The first tradeoff is known as PACELC [76], which advances the CAP theorem (shuffling the acronym) and adding: *Else Latency or Consistency*. The novelty of PACELC is considering the case when the system is not partitioned, stressing on the tradeoff that arises between latency and consistency.<sup>3</sup> A corollary to PACELC restricts the tradeoff to non-commutative transactions [77]. The observation that enables this corollary is that if two non-conflicting (commutative) transactions are performed in two different partitions, the overall consistency of the system is not compromised. This is an important restriction because, if transactions could be defined to be always commutative (which is impossible in a partitioned system), then an always consistent and available distributed system could be designed. An interesting consequence is that, if there are only few non-commutative transactions, all others can be executed in parallel to improve performance without sacrificing consistency. The idea to define partitions of the systems that can process subsets of commutative (non-conflicting) transactions is also known as *Sharding*, and empowers some of the most scalable permissionless blockchain solutions, e.g., *Chainspace* and *Omniledger* [78, 79].

### 3.2 The Blockchain Trilemma

The second tradeoff is known as the “*blockchain trilemma*”, illustrated in Figure 6, which is essentially the reformulation of the PACELC theorem for the blockchain domain. In particular, the trilemma illustrates the conjecture that a blockchain system cannot exhibit maximum decentralization, security and scalability (performance) at the same time.

Limited by the trilemma, an IoT developer willing to improve the network scalability may chose a consensus protocol less expensive than PoW or reduce the mining difficulty to

<sup>3</sup> Recall that the Bitcoin protocol enforces consistency introducing, by design, an average latency of 10 minutes per block, and further recommends to consider a block as “unconfirmed” until it becomes 6 blocks deep (Section 2.2).

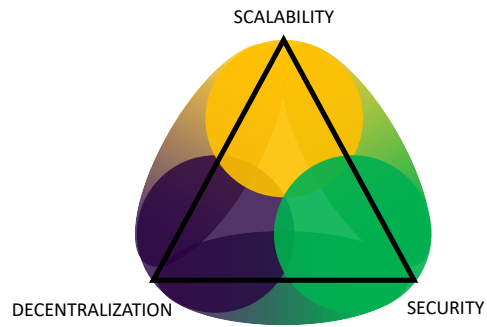


Figure 6: The blockchain trilemma is illustrated by a triangle like the one in this figure where one cannot draw a single point that is close to all the triangle corners, meaning that a trade-off among the three properties must be chosen [80].

speed up the block generation rate. However, this would compromise the security, since less computing power becomes sufficient to perform a successful attack. Another strategy could be to change the trust model, for example, restricting the access to the blockchain only to trusted, registered users. This is fundamentally the strategy adopted with permissioned ledgers, in which a central registrar is introduced to authenticate users, but in this case decentralization is traded for a performance gain. Again, a trade-off must be chosen, as the trilemma warns us that no consensus protocol can ensure full security, decentralization and scalability [80].

An IoT developer should therefore choose a consensus protocol and a blockchain-based system only after having clearly identified the application requirements, choosing the most appropriate trade-off. A brief review of consensus protocols is reported in in Section 3.3 to ease the selection of a consensus protocol for the IoT.

### 3.3 Brief Review of Consensus Protocols

Consensus protocols are commonly partitioned into two broad families: Voting and lottery-based protocols. A third category is introduced in this short review to classify those protocols that elude this general dichotomy. Table 1 is reported while concluding this section and compares the surveyed consensus mechanisms in terms of scalability, security and decentralization.

#### 3.3.1 Voting based Protocols

Consensus protocols address the metaphor of the *Byzantine Generals problem* [81], i.e., the challenge for an ensemble of commanders to coordinate to perform a successful attack despite the potential betrayal of messengers, where betrayals model nodes/links failures and malicious attacks. In classical solutions to this challenge, there is a node playing the role of “leader”, which multicast a transaction to all other nodes, which in turn try to simulate the transaction locally and send back acknowledgments about the status of the operation. If the local transaction simulation completes with success, then the node replies with an acknowledgment suggesting to commit, otherwise to abort. Acknowledgments can be seen as votes. As soon as the leader collects a trusted majority of votes for either commit or abort, then the decision is taken and broadcast back to all nodes.

*Leader-based and PBFT variants* The most popular protocol implementing this scheme is the Practical Byzantine Fault Tolerance (PBFT) protocol [82]. It tolerates up to  $f$  malicious entities in a system with  $3f + 1$  nodes, this is obtained requiring a quorum threshold of  $2f + 1$  that ensures a majority of working/honest nodes. Many variants exist, they vary in terms of number of voting phases, mechanism to elect or change the leader and quorum thresholds. Well known examples of consensus protocols based on voting are the classic 2- and 3-PC [83] or PAXOS [84], and many other more recent proposals [85–90].

Leader-based and PBFT variants are protocols that typically block (i.e., stop the decision process) when communication is asynchronous, where the keyword *asynchronous* is used to represent all the reasons for a communication delay (the message arriving too late) or failure, such as unreliable channels, faulty or even malicious nodes. By blocking, safety is ensured at the cost of an increased latency, while progress (liveness) is granted only when the network recovers from the transient asynchronous phase.

The potential blocks due to the failure of the leader are apparently inescapable. A unique leader must be part of the design; otherwise, a unique order of transactions cannot be defined. This observation leads to the conjecture that the distributed consensus problem is equivalent to the problem of unique leader election [91]. In fact, leader election is harder than consensus.<sup>4</sup> Unfortunately, the need of a leader comes with severe consequences [93, 94]. First of all, a single point of failure is introduced, so that it is sufficient to attack the leader via Distributed Denial of Service (DDoS) to block consensus. Secondly, the decisions taken by a single leader are not validated by any peer: having a leader becomes therefore a concern for the protocol fairness.

*Pure Voting* In a pure voting system, each node sends its vote to all others, letting everybody perform the counting operations locally. Running a voting phase in a pure voting system with  $N$  nodes means that each of the  $N$  nodes will send a vote to all other  $N - 1$  nodes, for an overall  $\mathcal{O}(N^2)$  communication complexity. Pure voting systems avoid single point of failure, but their quadratic complexity prevents their deployment at the scale of an IoT network.

*Federated Voting and Sharding* Pure voting systems are impractical while leader based protocols, albeit efficient, introduce a single point of failure. An hybrid approach can mitigate these issues. With Federated Voting or Sharding the network is partitioned and the consensus protocols are decomposed in smaller subproblems, solved within the federated enclaves (shards). The Stellar protocol [95] is a well-known representative of this class of protocols. With Stellar, each node independently selects a set of trusted nodes (e.g., trusted because of neighboring relationships) sets its own quorum threshold, then, applying a recursive principle of message passing and quorum overlapping, consensus can be achieved with a gossip protocol [96] in the whole federated network. Gossip protocols normally require to work in cycles to guarantee convergence, introducing once more the problem of asynchronous networks.

*Virtual Voting* Gossip algorithms are also the key feature of *virtual voting* protocols, such as the hashgraph consensus algorithm [97]. The idea is to let nodes spread transactions epidemically, attaching metadata about which received transactions have been critical to trigger a new transaction. At convergence, each node will own the full causal relationship between transactions, thus each node will be able to causally sort them, solving the consensus problem on their order. A causal consistency criteria [98] is well known to be weaker than sequential consistency [99], but improves on latency by better supporting non-blocking recording of transactions [100]. The keyword *virtual* is used because votes are not explicitly broadcast in an all-to-all fashion as described for pure voting systems, still, all nodes implicitly acquire sufficient knowledge to sort transactions.

### 3.3.2 Lotteries and Proofs

So far the consensus problem has been solved by voting, with votes used to elect a leader that imposes his decision or to form a majority in favor of a particular action. Another way to address the consensus problem is to organize a lottery, and the lottery winner becomes the leader. The winner must provide a sort of winning ticket, a “proof” to be shown to claim the consensus. All of the consensus protocols that resemble a lottery game are associated with a specific proof that is required to lead the consensus.

*Proof of Work (PoW)* For instance, in Section 2.2 the PoW has been introduced and the nonce is the winning ticket that miners need to show to propose a block, imposing their order of transactions. Albeit effective to achieve consensus, the drawbacks of PoW are evident. First of all, it slows transactions speed, but also results in a considerable waste of computing power and lack of fairness, as long as miners are free to include the transactions they prefer in the block they are trying to forge, thus can deliberately ignore the transactions of competitors.

*Proof of Stake (PoS)* The PoS is an energy-aware alternative to PoW that relies on economical rationality to achieve consensus. In PoS, a randomized process selects a leader, and the key property of the random process is to bias those entities that own more cryptocurrencies, or whatever resource is at stake. The reasoning is that the owners of many cryptocurrencies (the richest stakeholders) have a vested interest in keeping the network working well and trusted, so that the system could be perceived as valuable, and the value of the owned cryptocurrencies is safeguarded and enhanced.

*Delegated Proof of Stake (DPoS)* A variant of PoS that recently became very popular is DPoS, which is implemented for example by EOS, Tron, Steem, and Bitshares, and outperforms all other consensus protocols in terms of scalability [101]. With DPoS stakeholders vote to elect delegates, and their votes are weighted according to the fraction of owned coins. Sometimes delegates need to show commitment with a deposit (escrow) that can be confiscated if they do not run the internal consensus protocol honestly. The result is that delegates are chosen according to an economic rational criterion, and given that delegates are few and trustable (they are committed and accountable), they can achieve consensus much faster.

<sup>4</sup> “The weakest failure detector needed to solve Election is stronger than the weakest failure detector needed to solve Consensus” [92].

Table 1: Comparison of Consensus Techniques

	<i>Transaction Rate</i>	<i>Network Scalability</i>	<i>Consensus Participants</i>	<i>Attacks</i>	<i>Centralization</i>
<b>Leader-based</b>	High	High $\sim \mathcal{O}(N)$	Registered nodes	Collusion of 1/3+1 nodes	Central coordinator
<b>Pure Voting</b>	Low	Low $\sim \mathcal{O}(N^2)$	Registered nodes	Collusion of 50%+1 nodes	Fully distributed
<b>Sharding</b>	Medium/high	Medium	Known neighbors	Colluded strategic minority	Federations
<b>Virtual Voting</b>	Medium/high	Medium	Known neighbors	Neighbors can cheat	Fully distributed
<b>PoW</b>	Low	Low	Anonymous	Attack with control on huge computing power	Fully distributed
<b>PoS</b>	Medium	Medium	Anonymous	Collusion of the richest	Fully distributed
<b>DPoS</b>	High	High	Elected Delegates	Collusion of delegates	Requires delegates
<b>Round-Robin</b>	High	High	Registered nodes	Block by any malicious user	Requires global synch
<b>node-to-node</b>	High	High	Known neighbors	Neighbors can cheat	Fully distributed

*Other Proofs* Many other lottery based protocols have been proposed in the last years. They all evolve around the concept of proving commitment either by showing to be willing to sacrifice resources, or by the fact that the user owns a considerable stake. For the sake of comprehensiveness, other known proofs are:

- Proof of Elapsed Time (PoET) [102]: where the sacrificed resource is the (random) time spent in a waiting queue;
- Proof of Importance (PoI)<sup>5</sup> and Proof of Networking (PoN) [103]: where the node commitment in the network is computed on the base of a different mix of metrics, including network topological information;
- Proof of Burn (PoB): a node must burn coins, sending them to a dead address, to gain the privilege of leading the consensus;
- Proof of Capacity (PoC): if memory is the main resource necessary to solve a cryptographical problem, then PoW becomes PoC, like in [104];
- Proof of Deposit (PoD): with PoS, nodes with “nothing at stake” can behave maliciously without any punishment. In Casper [105], entities have to deposit some coins that are confiscated in case of malicious activities.

### 3.3.3 Other Approaches

*Round-Robin* A straight-forward mechanism to address the distributed consensus protocol is to let all nodes succeed each other, in consecutive turns, as leader. Full trust among nodes is required to fairly run the successions procedure, which can be blocked indefinitely by any malicious participant.

*Node-to-node Consensus* The *node-to-node* keyword is used to indicate two or more mutually trusting neighbors that agree to privately perform transactions. Usually these neighbors open a fast-payment channel to handle their frequent private transactions, so that they avoid paying the multiple fees that would incur if a public blockchain were used instead. Node-to-node transactions are not visible on a main blockchain, so they are said to be *off-chain* [106, 107]. Infrequently, the two nodes close their channel issuing a closing transaction reported on a

<sup>5</sup> This concept was introduced in NEM P2P cryptocurrency <https://docs.nem.io/ja/gen-info/what-is-poi>, but never published in a scientific location to the best of our knowledge.

main blockchain. This closing transaction may represent the balance of thousands or more off-chain transactions: the off-chain transactions rate can hugely enhance the overall network transaction rate.

## 4 Towards a Blockchain Definition

We have discussed the structure of the blockchain (Section 2) and distributed consensus protocols (Section 3), stressing on the limits and trade-offs inherent to the blockchain technology. However, browsing the literature related to blockchain applications makes the blockchain seem as an almost universal, limitless technology. The application range for the blockchain, in fact, seems to be so broad to include: Supply Chain Management [1–4], E-Voting [5–12], Smart-Grid [13–20], Healthcare [21–24], Banking [25, 26], Smart Cities [27–29] and even Vehicular [30–43] and Drone [44–48] Networks, going way beyond the original cryptocurrencies such as Bitcoin and Ethereum. We argue that this apparent universality of the blockchain is rooted in the ambiguity of the *blockchain* term itself. This ambiguity complicates clarification, since the blockchain can have different roles for the IoT depending on the different possible —if not improper— definitions.

To unriddle the possible usage of the blockchain we first need to formulate a connotative and precisating definition for the term “blockchain.” To this end, we compare the most popular platforms commonly considered as emblematic blockchains with standard Data Bases (DBs), looking for the distinguishing features that will constitute our blockchain definition. Thanks to this definition we can elucidate the conditions for a proper use of a blockchain, and we also provide guidelines for deciding when to avoid it and instead prefer to rely on traditional solutions.

### 4.1 Blockchain vs. Traditional Technologies

Figure 7 aids the discussion about the technological advantages and disadvantages of the competing technologies for the implementation of a Shared Ledger. The first two are traditional DB technologies, that will be compared with the still undefined concept of “Classic Blockchains.” Our intention is to capture under this concept all those platforms (such as Ethereum and Monero [108]) that preserved the distinctive features introduced in history for the first time by Bitcoin, marking a new era



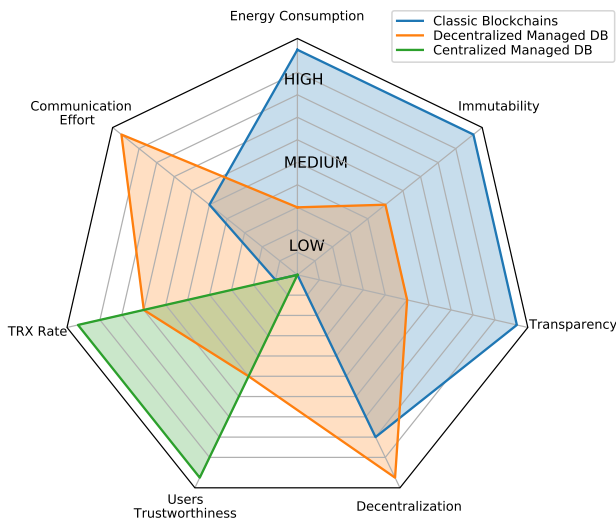


Figure 7: Multidimensional comparison of the blockchain with traditional Shared Ledger technologies: i.e., centralized and decentralized managed DBs.

for the Shared Ledger technologies.

**Centrally Managed DBs** A centrally managed DB is maintained by a central administrator, such as a trusted employee or a single company in charge of keeping the DB well maintained. The recorded data can be shared among various clients upon request. The central manager can, at his own discretion, authorize or deny the access to the DB. According to the described paradigm a centrally managed DB represents a possible implementation of a Shared Ledger.

The greatest advantage of one such implementation is the high level of efficiency in terms of transaction rate, communication effort and power consumption. In fact, an administrator can exploit the decades refined technology of commercial DBs to achieve maximum performance at a low power consumption. The administrator works autonomously, so it also avoids the communication efforts of a consensus protocol that would become necessary to coordinate more DB maintainers. If advantages are many, disadvantages are numerous too. For example, the trust in the administrator must be absolute because this administrator can in principle tamper, censor or even resell users data. A centrally managed DB is not considered transparent as well, because nobody controls nor validates the admin operations. Similarly it cannot even be considered immutable, as the admin is free to delete data.

**Distributedly Managed DBs** Distributedly managed DBs, i.e., DBs cooperatively maintained by a group of administrators, represented the only option to implement a decentralized Shared Ledger before the rise of blockchains. Redundant DB copies are introduced: nodes chose and run a consensus protocol to agree on writing operations, enforcing this way a consistency model [99, 109]. This distributed architecture provides a varying degree of tolerance to failures which depends on the strength of the consensus protocol and on the number of redundant DB copies. The price paid by distributed DBs for decentralization is the increased coordination effort necessary to run the consensus protocol, that also slows down the transaction rate. A distributed DB is harder to tamper compared to a

centralized one, since an attacker must corrupt more nodes. All write operations are validated by a quorum of peers: this mechanism enhances transparency as no absolute trust in the admin is required anymore. Nonetheless, the system is secure only if a majority of peers is honest. The maintainers of the distributed DB are free to record data in any data structure (not necessarily a block-chain), provided that the application requirements are enforced by other factors, e.g., through rules of the internal consensus protocol.

**Classic Blockchains** Iconic blockchain platforms are Bitcoin (described in Section 2) and Ethereum. Both are based on the PoW, precisely as many other popular PoW-based blockchains<sup>6</sup> that, together, account for more than 90% of the total market capitalization of existing digital cryptocurrencies [111, 112]. Some authors refer to these public, permissionless, PoW-based blockchains as to *classic blockchains* [61].

Classic blockchains turns out to be a particular case of decentralized DB where transparency and immutability are constitutional and brought to their extremes. The only data structure used in a classic blockchain is, unquestionably, a block-chain, i.e., a special linked list characterized by cryptographic links, and blocks of transactions as items of the list. In a blockchain, data can only be appended and it is never deleted or modified. All append operations are public and transparent, so that the validity of all transactions can be verified at anytime by any peer. A classic blockchain is open to any anonymous user, therefore a very strong consensus protocol is necessary to safeguard the ledger. This leads to the well known drawbacks: slow transactions rate, high latency, and huge power consumption. Albeit slow, strong consensus mechanisms adopted in public blockchains allow the removal of trusted authorities and enable transactions also among anonymous, untrustworthy users.

## 4.2 Connotative Definition of Blockchain

The analysis of the competing technologies for the implementation of a Shared Ledger (Section 4.1) suggests what are the distinctive characteristics of a blockchain that distinguish it from all other Distributed Ledger Technologies (DLTs). We condense these characteristics in the following definition of the term “blockchain”:

### Def. 4.1: Characteristics of a Classic Blockchain

1. OPENNESS TO ANONYMOUS USERS
2. FULL & PUBLIC HISTORY OF TRANSACTIONS
3. STRONG DISTRIBUTED CONSENSUS PROTOCOL

The OPENNESS TO ANONYMOUS USERS is the first, essential feature of a blockchain. The blockchain ability to preserve the privacy of users more than what is done by banks or by other centralized implementations of a Shared Ledger comes ultimately from the anonymity of users. The openness to anonymous users is also fundamental for making blockchains decentralized. If users had to be identified, a centralized trusted

<sup>6</sup> Examples of other famous PoW-based cryptocurrencies are Bitcoin-Cash, Litecoin, Namecoin, Dogecoin, Primecoin, Auroracoin, Monero, Ethereum-Classic and Zcash. For a more complete list of cryptocurrencies the reader can refer to [110].

registrar —potentially discriminatory— would become necessary, compromising the ledger decentralization. The openness to anonymous users is thus constitutional for a blockchain, but introduces also a new problem about the disputation of transactions, because it is not possible to prosecute an anonymous, untraceable user in case of fraud: users must accept that transactions are, de facto, indisputable.

In the trustless scenario made of anonymous users, one can accept an indisputable transaction only if it is empowered to perform, on its own, a complete check of validity of any transaction at any time. Involving a trusted authority, such as a bank, this user could trust the private and opaque internal ledger of this bank to manage transactions, but removing trusted intermediaries implies the necessity of keeping a complete record of all transactions on a ledger open to the public, otherwise the distributed validation of transactions becomes impossible. As seen in Section 2.1, the solution offered by blockchains is to record the PUBLIC & FULL HISTORY OF TRANSACTIONS, so that anyone can verify that no previous transactions in the whole history already spent the resources being transacted.

Despite the availability of a public and complete ledger, the validation of a transaction can be still falsified by an attacker that manipulated the history of this transaction, tampering the blockchain for gaining a personal advantage. Therefore the ledger (i.e., the blockchain) must be safeguarded by a STRONG DISTRIBUTED CONSENSUS PROTOCOL, otherwise nobody would trust the system. This is why a mechanism like the PoW that, as explained in Section 2.5, makes historical blocks immutable and is mathematically secure regardless of any trust assumption on users, is a distinctive element of blockchains. This kind of immutability is so fundamental for the concept of blockchain that [sic] *for cryptocurrency activists and blockchain proponents even simply questioning the immutable nature of blockchain is tantamount to heresy* [113].

To recap: a blockchain is designed specifically to guarantee full memory of all transactions open to any anonymous user to enable the distributed validation of transactions avoiding trusted intermediaries. But the blockchain alone is meaningless without a mechanism that safeguard the historical records of transactions from tampering attacks, this is why a STRONG DISTRIBUTED CONSENSUS PROTOCOL becomes essential. The arguments we used to justify our definition of blockchain are concisely summarized in Highlight 4.1.

#### Highlight 4.1: Arguments supporting Definition 4.1

Users Anonymity  $\Rightarrow$  non-disputable Transactions;

If the Ledger is  $\_$  then New Transactions are  $\_$  :

- Private  $\vee$  Partial  $\Rightarrow$  *unverifiable*
- Public  $\wedge$  Full  $\Rightarrow$  *verifiable*

A Strong Consensus makes the Ledger immutable, protecting it from falsification.

#### 4.2.1 Comparison with other definitions

A first definition of blockchain, in computer science, can be restricted to the simple data-structure made of blocks of information chained by hash-pointers, known in the literature since the '70s [114, 115]. However, we believe that the introduction of Bitcoin and Ethereum enlarged the meaning of the term

blockchain. As a matter of fact, Iansiti and Lakhani propose a wider definition which is the following: “[The] blockchain is an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way” [116].

This definition highlights the blockchain operational purpose as distributed ledger, distinguished from other traditional ledgers because of its Openness, Verifiability and Immutability (permanent records) properties. Our characterization highlights these same features but it does so stressing on the constitutional elements that connote a blockchain as an open, verifiable and immutable ledger technology. Definition 4.1, in fact, characterizes the blockchain as a technology *open* to any anonymous user, *verifiable* thanks to the complete and public recording of all transactions, and as much *immutable* as possible by reason of a strong distributed consensus protocol. Our definition, compared to previous ones, is not axiomatic: rather, acknowledges the blockchain as an open, verifiable and immutable technology deriving these properties from the three, inseparable elements that together constitute the essence of a blockchain.

Our definition results to be more restrictive, as it reserves the epithet “blockchain” only to those platforms that fully reflect the blockchain specificity described by Definition 4.1. The various platforms usually mentioned as blockchain without actually complying with Definition 4.1 are criticized in Section 4.3.

Definition 4.1 also serves our clarification purpose, being the base for claiming that *the blockchain is not a universal technology*, but it rather has precise characteristics advantageous only for a limited number of applications. The many applications proposed in the recent literature that exhibit features in contrast with Definition 4.1 are criticized in Section 5.

### 4.3 Permissioned Ledgers are Blockchains?

Definition 4.1 raises a question: since permissioned ledgers are not openly verifiable, nor safeguarded by a strong consensus protocol, shall we call them blockchains?

**Not an Open, Decentralized, Verifiable Technology** By definition, in permissioned ledgers the access is restricted only to permissioned users. A central, trusted registrar responsible for the identification of users and for granting permissions must therefore exist. Moreover, permissioned ledgers are typically used by enterprises to record business-critical transactions, that consequently are kept confidential and cannot be verified by an external agent. For these reasons, permissioned ledgers are not a truly decentralized nor an open technology.

**Less Immutable means less Secure** A trust model with registered and permissioned users is certainly safer than one where the ledger is open to any anonymous user. Strengthened by stronger trust assumptions, permissioned ledgers usually abandon the secure but power-hungry PoW replacing it with more traditional, efficient consensus protocols. However, this way they return to be vulnerable to traditional attacks led by the “simple” — i.e., “inexpensive”, not discouraged by any costly sacrifice— collusion of a majority of users. Permissioned ledgers are therefore less immutable and less secure than iconic blockchains such as Bitcoin or Ethereum, that instead are not affected by this vulnerability.

**Permissioned Platforms are Traditional Ledgers** Permissioned platforms seem to be not much different from traditional ledgers that existed also before Bitcoin [117], as they are empowered by traditional consensus protocols and their trust model still depends on a central authority, while permissionless blockchains such as Bitcoin revolutionized the state-of-the-art. From an historical perspective permissioned platforms represents therefore only a technological sophistication of the older, traditional DLTs.

For these reasons, from now on we will consider permissioned platforms as belonging to the broader class of traditional DB-based ledgers, rather than blockchain representatives. Figure 8 depicts our vision of the landscape of Shared Ledger technologies, with the blockchain positioned according to Definition 4.1 as provided in Section 4.2.

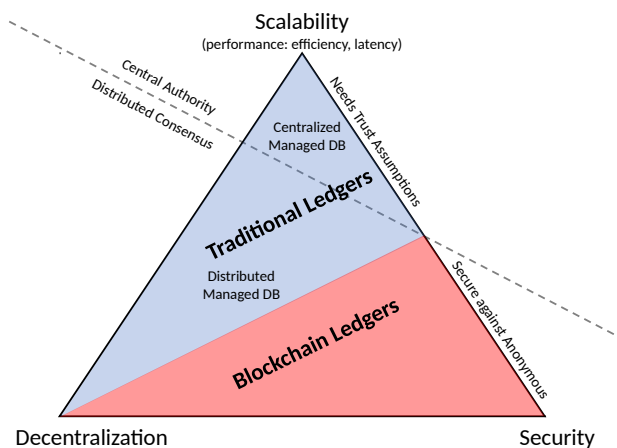


Figure 8: Position of the Blockchain in the landscape of the Shared Ledger technologies.

#### 4.4 Proof of Work or Proof of Stake?

The transition from PoW to PoS planned by many popular blockchain systems (in primis Ethereum) to stop wasting computing power can compromise the blockchain characteristics? Many would be the reasons to dismiss the PoW:

- Enormous power consumption (Section 2.7 and [59, 73]);
- Modest transaction rate and high latency [118, 119], if compared to traditional Byzantine-Fault-Tolerant alternative protocols [120];
- Tendency for the computing power to consolidate under the control of few large mining pools<sup>7</sup> [122, 123]. Moreover, under PoW it may be advantageous for few powerful users to collude behaving as “*selfish miners*,” again damaging the decentralization level of the network [124, 125].

For these reasons the transition to the PoS may be justified, however, also the PoS has been criticized:

- The PoS design deliberately priorities the richest stakeholders, these last tend to accumulate voting power damaging the network decentralization. This tendency is usually condensed in the motto “*the richer get richer*” [126].

<sup>7</sup> The popular blockchain.com website reports the hashrate distribution of the largest Bitcoin mining pools [121], with the 12 largest of them controlling almost the 80% of the network computing power. Notice, however, that pools are consortia that aggregate together multiple miners, so they still achieve a certain degree of decentralization.

- While there exist rigorous studies on the convergence and on the byzantine-tolerance of both traditional voting protocols and of PoW-based blockchains [71], the economics of stake-based systems suggest that their equilibrium is not always granted [127, 128]. This raises the concern that, like with real markets, stake-based protocols will lead to unstable systems subject to market failures and bubbles [129].
- A PoS-based blockchain is reversible by means of long-range or stake-bleeding attacks [130, 131], thus its immutability is considered questionable.

In light of the discussions on the limits of PoW and PoS we claim that, essentially, they both empower a *census suffrage* system. In the case of PoW only rich users that can afford the sophisticated mining equipment can participate in the protocol. In PoS a similar restriction on voting by census is directly embedded in the protocol, with the remarkable advantage of saving a huge amount of energy, but with the risk of long term instabilities. There is, however, a key difference: while the acquisition of computing power is subject to natural factors such as the cost of electricity, the value fluctuations of a PoS-system only depends on speculative mechanisms. So while with PoW it is a mathematical fact that the cost of an attack increases exponentially with the number of blocks to be changed, and this cost is bounded to a physically enormous amount of energy, the same cost for an attacker of a PoS system is unpredictable, because the cost of the “value-at-stake” for an attacker is not bounded to any external factor.

This position paper concludes that a PoW-based system is more stable and predictable than a PoS-based one, and therefore suggests that a platform that candidates itself to be the most secure one must be PoW-based.

## 5 Do you need a Blockchain? Avoid Common Pitfalls!

A reader that accepts the blockchain defined as the open, verifiable, and immutable Shared Ledger technology par excellence, immutable by reason of a powerful consensus protocol, should also acknowledge it as extremely inefficient [73, 132]. For this reason, we recommend to use the blockchain technology only when needed, opting for a different technology whenever possible, especially for the IoT. For example, a traditional ledger is preferable when the access is restricted to registered users, or when data must be kept confidential, or when strong trust assumptions are given, which makes the strong consensus required by the blockchain an overkill. The above considerations are illustrated in Figure 9, which extends the tradition started by Peck and Wüst [133, 134] to provide a guided path to clearly recognize when a blockchain is not the right choice for an application. The one provided here distinguishes itself from previous ones [135] for its limited scope, i.e., highlighting the cases when the blockchain is *not needed*.

Figure 9 also highlights, implicitly, those recurrent abuses of the blockchain in applications whose requirements conflict with the essential blockchain characteristics. In the rest of this section we list these recurrent pitfalls to substantiate our critical analysis the blockchain applications.

*Pitfall 5.1: Using the blockchain in presence of strong assumptions on trust.*

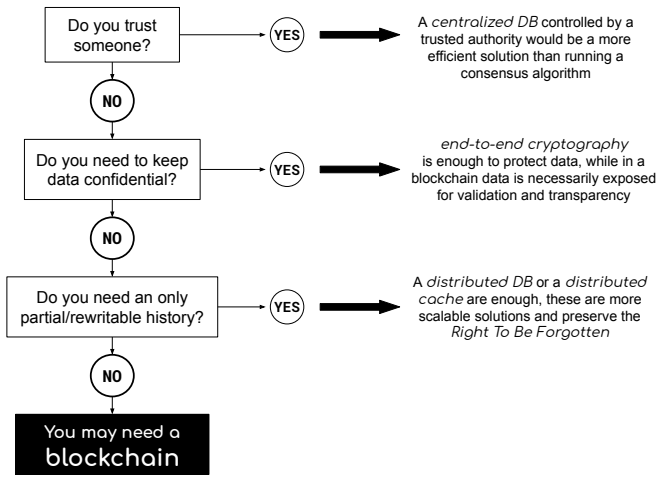


Figure 9: Application requirements and ledger technology: Aid to decision.

Morgen E. Peck first identified this pitfall by analyzing the application of the blockchain to voting [133], nonetheless, many are the works that propose blockchain-based voting platforms [5–12]. We note that a blockchain as defined by Definition 4.1 contrasts the needs of voting because, although it is true that the voter’s identity should be kept secret, still users cannot be anonymous. Their identity must be in fact uniquely determined to ensure uniform eligibility, namely, nobody should be able to cast multiple votes (Sybil attack), “hence an identity provider is required one way or another” [136]. This identity provider must be necessarily a centralized accredited institution unless, ad absurdum, someone accept voters providing IDs issued by unofficial distributed agencies (fake documents!). Double-voting is thus different from double-spending and is a problem whose solution imposes the existence of a central authority: this last cannot be decentralized not even advocating a blockchain.

In general, the existence of a trusted third party (such as an identity provider) or the assumption of mutual trust in a distributed network are considered strong assumptions. If a trusted third party exists, then this party can play the role of the central coordinator, and by hypothesis users can rely on it to keep a consistent DB without going through the overhead generated by a consensus algorithm. From the perspective of the blockchain trilemma, the need of a trusted authority is equivalent to trade decentralization for scalability. Clearly, under this assumption the sacrifices that come with the blockchain are not necessary. Similarly, assuming mutual trust among nodes, there is no need to trade performance for security.

*Pitfall 5.2: Proposing a fast blockchain.* Fast blockchains of all kinds have been proposed, especially for business applications, circumventing the limits of the PoW by promoting alternative protocols. Some popular platforms advertised as secure and fast include the HyperLedger Fabric, HyperLedger Sawtooth, Ripple, Amazon Managed Blockchain and Azure Blockchain, followed by many others. All these systems are collectively called Blockchain-as-a-Service (BaaS) Platforms [137]. However, all of them inevitably sacrifice some specific features of the blockchain in an effort to remove those that are unsuitable for the enterprise. We raise the concern that, according to the blockchain trilemma, the enhanced performance [138] granted by these platforms can be achieved only

reducing, partially, either the decentralization or the security of what this paper defined to be a blockchain. Two exemplary fast platforms will be now studied to highlight the great differences with classic blockchains such as Bitcoin and Ethereum. We stress on these differences to justify, once more, the restrictive Definition 4.1 proposed by this paper. Not calling “blockchain” these fast ledgers is a terminological choice only but we not discourage their use, rather, we promote their adoption when a blockchain is not needed, especially in the IoT (see Figure 9 again).

The official documentation of **HyperLedger Fabric** [139] describes its core design. It also highlights that the essential characteristics of public permissionless blockchains, such as being *public networks, open to anyone, where participants interact anonymously* are problematic for enterprises, especially because *the identity of the participants is a hard requirement* for enterprises to comply with legal obligations such as Know-Your-Customer (KYC) and Anti-Money Laundering (AML) financial regulations. This leads to the identification of a list of requirements for enterprises:

- Participants must be identified/identifiable.
- Networks need to be permissioned.
- High transaction throughput performance.
- Low latency of transaction confirmation.
- Privacy and confidentiality of transactions and data pertaining to business transactions.

As the HyperLedger Fabric “has been designed for enterprise use from the outset” these requirements are all mandatory. The following short analysis of HyperLedger Fabric (Fabric, for short) explains how it supports the listed features.

A membership service associates entities in the network with cryptographic identities. Fabric enables Privacy and Confidentiality through its Channels architecture, where Channels are defined as *private “subnet” of communication between two or more specific network members, for the purpose of conducting private and confidential transactions* [140], and through *private data* [141]. The Fabric documentation also reports this noteworthy consideration: *By relying on the identities of the participants, a permissioned blockchain can use more traditional crash fault tolerant (CFT) or byzantine fault tolerant (BFT) consensus protocols that do not require costly mining.* The consensus protocols supported by Fabric are indeed traditional consensus protocols of this kind. In particular, the leader-based voting consensus protocol *Raft* [85] is the only non-deprecated protocol for deployments of Fabric. With these features, that set Fabric apart from the blockchain, Fabric can reach 20 kTPS [138], outperforming known blockchains by 4 orders of magnitude.

Observing Fabric we notice the use of more efficient and traditional consensus mechanisms, resulting in greater performance compared to classic blockchains. Moreover, its security derives from a traditional access control (permissions) but not from a strong consensus mechanisms. The resulting ledger is not open and is vulnerable to collusive attacks lead by a majority of users not defused by any deterrent cost such as the CPU energy implied by the PoW. As such, Fabric is more similar to traditional ledgers than to classic blockchain systems like Bitcoin. It is in fact a decentralized platform customized for the

enterprise. That is why we include it in the class of more traditional Shared Ledgers [142].

**HyperLedger Sawtooth** [143] is an extension of HyperLedger that explicitly requires the Intel SGX framework. Sawtooth promotes the Proof of Elapsed Time (PoET) as a *solution to the Byzantine Generals Problem that utilizes a “trusted execution environment” to improve on the efficiency of present solutions such as Proof-of-Work ... and assumes the use of Intel SGX as the trusted execution environment* [144]. The idea behind PoET is the following. A random waiting time is distributed to all nodes competing to become the next block miner. When the waiting time expires, the node proves that it waited by providing the PoET generated by its Intel chip. The first node that exhibits a valid PoET is elected as block-miner. This protocol is much more energy efficient since the Intel chip consumes much less than a Bitcoin miner to generate a PoET. However, in this protocol users must trust the server distributing random waiting times and must also trust the proprietary Intel SGX technology, (which has been already attacked successfully multiple times because of its internal architectural flaws [145]). Moreover, if SGX chips are reasonably affordable it becomes possible for anyone to get many of them that, together, will consume a lot of power for increasing the chances to be elected as block-miner, falling back to a Bitcoin scenario.

In general, all *fast* or *low-energy* proofs facilitate attacks, so that mining-difficulty must be artificially kept high (as described for Bitcoin in Section 2.4) to ensure high level of security. The trilemma warns us to beware of fast consensus protocols. They should only be run in centralized platforms, hence in private, non transparent organizations. Otherwise, they are prone to be easily attacked. Therefore, applications implemented on top of BaaS platforms cannot be as secure and transparent as if implemented on top of what we defined to be a blockchain.

*Pitfall 5.3: Validating sensory data through a blockchain.*

This is a common pitfall besetting, for example, all those who choose the blockchain for supply chain management, a quintessential type of IoT application relying on sensor readings and other “things” from the physical world [1–4]. One is lured into using the blockchain for it makes handovers among intermediate dealers manifest, from the producer to the final customer. However, it cannot ensure that the traded goods have been transported correctly. As observed by Wüst et al., the problem with the supply chain lies in the *trust of sensors* or, in other terms, in the way trusted information is acquired from the real world [134]. For example, a malicious truck company that wants to cut costs of transport of refrigerated food can claim its trustworthiness showing compliance with the laws by installing “trusted” thermometers with GPS. The company can then cheat installing the thermometer in a little empty fridge traveling on the truck together with the rest of the (not refrigerated) load. In general, the “perception layer” of the IoT is the most vulnerable to attacks [146, 147], so that IoT devices (potentially deployed outdoor without supervision) must implement in-hardware mechanisms to be secured. Still no blockchain will ever be able to prevent all possible physical sabotages of sensors.

We claim that sensors cannot be trusted not just because they can be compromised, but also because of the inescapable *uncertainty of measurements*, independently on the source—whether benign or malicious—of the uncertainty. Consider, for exam-

ple, a smart contract used to buy train tickets that embeds in its business logic the automatic refund for travelers in the case of train delays above 30 minutes. It is not hard to imagine that the rail company may cheat by reporting (false) delays that are lower than 30 minutes to avoid paying refunds. All applications that rely on measurements, taken by any kind of sensors or IoT device, at some stage must trust either the centralized company controlling that sensor or a consortium that collected the measurement. When this happens we talk about *oracles* [148] that must be introduced to obviate the trust problem on sensor by providing trusted information services [149]. However, these oracles are either centralized trusted authorities (see Pitfall 5.1), or systems that require distributed consensus, which reintroduce all the issues and trade-offs discussed in Section 3.2. With oracles the blockchain loses its meaning as a tool to implement distributed trust.

*Pitfall 5.4: Proposing a blockchain-based approach for confidentiality.*

Confidentiality, namely, providing data secrecy, is critical for many applications. There is no reason to publish and record confidential data on a public blockchain as confidentiality is in clear contrast with a key characteristic of blockchains: Transparency, which is given by the sum of the blockchain openness and completeness. Works that advocate the use of the blockchain for keeping user data confidential include [22, 150–155]. We note that since confidentiality contrasts with the public nature of blockchains, a careful justification of design choices is necessary.

*Pitfall 5.5: Storing sensitive information on the blockchain.*

Registering user credentials and account information on a blockchain is an irreversible operation, as data cannot be deleted. Services implemented on top of a blockchain will not be able to delete user data; not even upon legitimate request. This could be an issue for a user that wants to abandon a service and also for authorities that need to enforce the “Right to be Forgotten,” which is a legal provision in several countries [113, 156–158].

*Pitfall 5.6: Verifying the authenticity of digital documents or real goods with a blockchain.*

Many proposals concern the use of the blockchain as a decentralized platform to store digitally signed documents, i.e., certificates [159–163]. One might be drawn to believe that, as it is part of a blockchain, that certificate is authentic. However, the authenticity of a certificate is guaranteed by its digital signature, which depends on a trusted authority external to the blockchain and not subject to the trust obtained via a consensus protocol. For example, let us consider the following certificate digitally signed by an institution  $I$  and recorded in a blockchain: “Alice passed exam  $E$  after attending the course provided by Institution  $I$  on date  $D$ .” The fact that this certificate belongs to a blockchain does not in any way prove that Alice really attended a course and passed an exam, and legitimate doubts can also arise about the timestamped date  $D$ . Trusting the authenticity of this document only relies on unconditionally trusting the issuer, namely, institution  $I$ . Overall, the blockchain can highlight how a series of operations is chronologically consistent, thus valid. However, for non transactional data like common certificates, patents and proofs of authorship, the blockchain cannot assist their authentication or validation.

This holds also for identity documents, as argued in Pitfall 5.1, and for real goods too. Somehow, in fact, many have been inspired by the blockchain immutability to guarantee the authenticity of real goods [1–3, 164–167]. The idea would be to associate any good with a digital identifier (e.g., a QR code) tracked by a blockchain so that a final customer receiving the good can use the attached identifier as a proof of authenticity. Unluckily, there is no mechanical tool to make objects of the real world unforgeable: For as much as the identifier on the blockchain is immutable (unforgeable), this is not true for the associated good, which can be physically replaced with another of lower quality. We stress that the *digital signature* is the technology for verifying the authenticity of transactions or digital documents but the blockchain alone, instead, cannot prove the authenticity of products or certificates.

*Pitfall 5.7: Forgetting the cost of mining.* One can think to the blockchain as the remover of banking fees. This is true as long as an incentive mechanism encourage validators in processing transactions, but turns to false as soon as this incentive mechanism terminates. Any blockchain project should describe a sound incentive mechanism for miners and consider transaction costs or it will be destined to failure, as no actors of the system will bear the cost of mining.

*Pitfall 5.8: Implementing a memoryless process on top of a blockchain.*

In a typical blockchain system the full chain of transactions must be recorded for validating new transactions. For those applications designed on top of Markovian assumptions, i.e., where only the knowledge of the *current state* of the system is necessary to make progress, using a blockchain will keep recording (possibly too many) unnecessary data. For example, there is no need for a smart home IoT application to keep memory of the full history of the temperature of a room to decide which heat source to activate or disable at the current time. Even accounting and billing does not require the entire history, but only a previous reading and recent invoicing.

*Pitfall 5.9: Claiming to jointly provide maximum security, decentralization and performance.*

Given the current state of science and technology, the proposals claiming the joint provision of maximum security, decentralization and performance violate the limits of distributed systems (Section 3.1). By blockchain trilemma they cannot truly work as promised. This is witnessed by the fact that, among the blockchain projects launched in the past few years, many have already failed [168] and some have been even denounced as Ponzi schemes [169].

## 6 Blockchain in Support of the IoT

Our exploration of the trade offs imposed by the blockchain trilemma (Figure 6) together with the clarification of the applications that, in light of our blockchain definition, result to be flawed, do not hint to any promising strategy for the *integration* of the blockchain within the IoT. However, we argue that the blockchain can still be used as an *external service* supporting the decentralized validation of IoT transactions, offering a complementary or alternative paradigm to centralized cloud services.

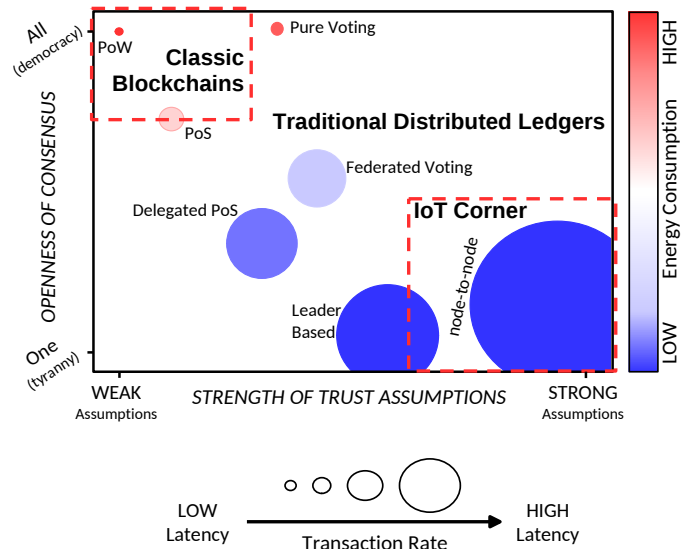


Figure 10: Bubblechart of relevant building blocks for Shared Ledgers. The diagram compares solutions as a function of trustworthiness of users (x-axis) and openness of the consensus protocol (y-axis). The color and the size of each bubble offer a quick indication of energy consumption and transaction rate of each consensus protocol, respectively.

To illustrate how the blockchain can successfully play as an external—not integrated—ledger for the IoT, we introduce the “bubblechart” of Figure 10, which draws a multidimensional overview of the consensus mechanisms surveyed in Section 3.3 according to the following four dimensions:

- The strength of the trust assumptions, which is inversely proportional to the degree of security (x-axis).
- The openness of consensus, an indicator of the degree of democracy, from one (tyranny) to all (power to the people) (y-axis).
- Energy consumption (color of each bubble: Red when high or blue if low).
- The transaction rate (size of each bubble: The larger the faster). The transaction rate can be considered also an indicator of transaction latency: Fixing the number of transactions registered with a single operation (block-size), a lower latency leads to a higher transaction rate.

The figure enriches the trilemma by breaking down the “scalability vertex” (Figure 6) into two distinct dimensions, i.e., energy consumption and transaction rate.

The ideal “blockchain-for-IoT” bubble would be blue and large (low-power and very performing) in the top-left (most decentralized and most secure) corner of the chart. Blockchain systems are naturally located in the top-left corner, characterized by being democratically open and secure despite weak trust assumptions, but also extremely slow and resource-hungry. The opposite corner is where IoT applications reside, with their scalability requirements, tight resource constraints, high global transaction rates. This corner also highlights that the ultimate participants are “things” rather than humans, thus affording a higher degree of trust, at least towards some other parts of the system. As such, Figure 10 pictorially conveys our crucial observation: Classic blockchains are in contrast with IoT requirements, which keeps the two realms well separated.

However, despite the lack of space for blockchain systems in the bottom-right corner of our bubblechart, going back to

Table 1 (last line), we identify *node-to-node consensus* as a means to build trust among operators/systems without established relations. The key word, here, is *node-to-node*, which is used to restrict the distributed consensus problem to few nodes, usually a couple although extensions to small numbers is efficiently conceivable. To settle a transaction it is sufficient for the transacting parties to agree on the transaction protocol, and this agreement can be reached privately by the two (or few more) parties in any fashion. What makes node-to-node consensus appealing for IoT is its efficient support of *local consensus*, which is natural for many IoT applications such as those with groups of sensors or a platoon of vehicles.

The number of different node-to-node consensus protocols is limited only by imagination. Specific transitive properties, i.e., how and to what extent if node A trusts node B and node B trusts node C, then node A can trust node C, can be defined to be applied to large clusters of trusted entities, ultimately leading to a network (the IoT itself in some sense) of diverse but interoperating “channels.” We inherit the term “channel” from the world of cryptocurrencies (*Networks of Payment Channels* [170–174]) and from that of communication networks, where a network is a set of channels interconnecting its nodes. Since IoT transactions are not necessarily monetary, in the rest of this work we use the generic term *Transaction Channels*. In the remainder of this section we describe the networks of Transaction Channels, and how they enable the external use of the blockchain for the IoT.

## 6.1 Networks of Transaction Channels

*Transaction Channels* are all those techniques used to group off-chain transactions between the same small group of users to speed them up and avoid paying multiple transaction fees. A Transaction Channel is therefore a node-to-node consensus protocol where the two transacting parties establish a fast “payment” method and agree to postpone the clearing of the transactions’ balance. Recording the status of the channel on the blockchain can be periodic or event-based. What is stored in the blockchain is a meaningful representation of the transactions’ history. For example, this could be the stochastic representation of a long-term distributed measure or the amount of energy exchanged in a smart grid.

The most notable implementation of Transaction Channels is the Lightning Network [106], which scales up the technique to a full network of such channels. This allow payments to be routed between remote end-points thanks to a dedicated routing protocol for payments, with the aim of providing a globally scalable mechanism for fast transactions. The Lightning Network is “Bitcoin oriented,” but the concept of a network of payment channels may become the transaction platform enabling a global market at the IoT scale. It also opens the way to thrilling research challenges such as bringing network science and expertise into the domain of transporting and routing payments within Payment Networks, as explored in [175]. Major open problems include addressing the depletion of channel capacity, especially for the most loaded nodes in the center of the network, developing enhanced centrality-aware routing strategies [176, 177], and rebalancing techniques [178–180].

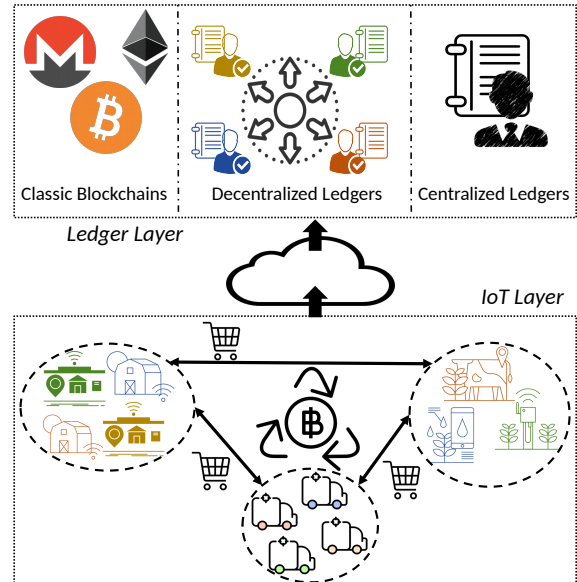


Figure 11: Different IoT clusters, made of devices managed either by private (home/enterprises) or public (institutional) entities perform most transactions locally, in the IoT layer. For interoperability purposes the different IoT clusters access intermediary platforms. Blockchain-based platforms can provide specific services at this level in line with their characteristic capabilities of granting security and immutability even in the absence of trust.

## 6.2 The Role of the Blockchain in the IoT

Figure 11 illustrates our vision of the IoT empowered by Networks of Transaction Channels deployed at the IoT layer. Here blockchains can play as supporting external ledgers, similarly to how the Bitcoin blockchain supports the recording of the channels status in the Lightning Network.

This vision stems from the observation that most IoT applications will have a *local* relevance. For example, domestic devices will intercommunicate mostly only over a house local network. Similarly, agricultural sensors for precision farming will mostly communicate with each other and convey the local information to a gateway controlled by the farmer. Industrial IoT often requires high levels of privacy and confidentiality, clearly in contrast with the open, immutable nature of a blockchain; vehicular networks and intelligent transportation systems may require transactions with latency smaller than a few milliseconds, and rates in the order of kTPS per vehicle, again in full contrast with the characteristics of blockchains. IoT transactions are local and normally lightweight in nature, therefore calling for local and lightweight solutions for the platform to support them. From time to time, separate IoT domains, platforms and applications may need to carry out and record transactions with a global, final, and immutable nature. At this level blockchains will play an important role, freeing IoT systems from the need to subscribe to a global, centralized, expensive, trust-based service whose security and reliability have well-known limitations. Using blockchains externally would therefore bring added value to the IoT domain, responding to its requirements of extending beyond local, context-limited applications when needed.

## 7 Conclusions

In this paper we argue that the blockchain is not the appropriate technology for securing the IoT, albeit it can bring added value as an external service. To support this claim we made some clarity around the very name “blockchain,” to dispel the many misunderstandings that hamper its usage and makes it appear as a universal—almost magic—technology.

For this purpose we explore consensus protocols, reviving those theorems that give rise to fundamental trade-offs such as the emblematic blockchain trilemma. We raise the concern that stake-based protocols fully rely on the rationality assumption of their users and, compared to traditional voting based protocols, lack of mathematical stability properties. This means that stake-based systems are prone to market failures and bubbles like real stock markets—a very dangerous risk. Moreover, we stress how the voting power has a tendency to consolidate in the hands of few great stakeholders with both PoS and PoW. For this reason, we suggest to consider the two as *census suffrage* mechanisms, with PoS preferable over PoW to reduce the energy consumption. However, the immutability of a PoS-blockchain is questionable, so a PoW-based one is considered more secure. In the landscape of the Shared Ledger technologies that we draw from the distributed system perspective of the IoT, we highlight the innovative and peculiar aspects of permissionless blockchains in contrast with permissioned ones, the latter turning out to be not so different from traditionally managed data bases. We conclude that the term “blockchain” should be reserved to those platforms characterized by: *i*) openness to anonymous users; *ii*) full and public history of transactions, and *iii*) safeguarded by a strong consensus protocol. This definition has far-reaching consequences. Above all, the strong consensus protocol requirement necessarily brings high resource consumption to counter the lack of trust between users, and imposes transactions rates and latency unacceptable for most IoT scenarios. As such, we stress that the blockchain is not a technology suitable for popular applications such as e-voting and supply chain management. Furthermore, to not violate the Right to Be Forgotten, the blockchain can hardly support Identity Management applications nor work as archive of certificates and other sensitive information.

In conclusion, we advocate using the blockchain only in those IoT scenarios where the transactions are supported by local, lightweight platforms whose consensus is tailored to the domain of application and the local context. We name these platforms “Transaction Channels.” These channels may (or may not, depending on the application) interact through aggregate, rare transactions to form a global network of Transaction Channels, which can be successfully based on the blockchain technology, freeing the IoT from the need to rely on global, centralized platforms to interact across diverse application, technology, and administrative domains.

## References

- [1] M. Montecchi, K. Plangger, and M. Etter, “It’s real, trust me! Establishing supply chain provenance using blockchain,” *Elsevier Business Horizons*, vol. 62, no. 3, pp. 283–293, may 2019.
- [2] D. Tse, B. Zhang, Y. Yang, C. Cheng, and H. Mu, “Blockchain Application in Food Supply Information Security,” in *Proc. of the IEEE Int. Conf. on Industrial Eng. and Eng. Manag. (IEEM’17)*, Singapore, dec 2017, pp. 1357–1361.
- [3] F. Tian, “An agri-food supply chain traceability system for China based on RFID and blockchain technology,” in *Proc. of the 13th IEEE Int. Conf. on Service Syst. and Service Manag. (ICSSSM’16)*, Kunming, China, jun 2016, pp. 1–6.
- [4] D. Miller, “Blockchain and the Internet of Things in the Industrial Sector,” *IEEE IT Professional*, vol. 20, no. 3, pp. 15–18, may 2018.
- [5] N. Kshetri and J. Voas, “Blockchain-Enabled E-Voting,” *IEEE Software*, vol. 35, no. 4, pp. 95–99, jul 2018.
- [6] A. Ben Ayed, “A Conceptual Secure Blockchain Based Electronic Voting System,” *AIRCC Int. J. of Netw. Security and Its Appl.*, vol. 9, no. 3, pp. 01–09, may 2017.
- [7] S. Bistarelli, M. Mantilacci, P. Santancini, and F. Santini, “An end-to-end voting-system based on bitcoin,” in *Proc. of the ACM Symp. on Applied Comput. (SAC’17)*, Marrakech, Morocco, apr 2017, pp. 1836–1841.
- [8] Y. Liu and Q. Wang, “An E-voting Protocol Based on Blockchain,” *IACR Cryptology ePrint Archive*, vol. 2017, no. 1043, 2017.
- [9] F. Sheer Hardwick, A. Gioulis, R. Naeem Akram, and K. Markantonakis, “E-Voting With Blockchain: An E-Voting Protocol with Decentralisation and Voter Privacy,” in *IEEE Int. Conf. on Internet of Things (iThings) and IEEE Green Comput. and Commun. (GreenCom) and IEEE Cyber, Physical and Social Comput. (CPSCom) and IEEE Smart Data (SmartData)*, Halifax, NS, Canada, jul 2018, pp. 1561–1567.
- [10] B. Wang, J. Sun, Y. He, D. Pang, and N. Lu, “Large-scale Election Based On Blockchain,” *Elsevier Procedia Computer Science*, vol. 129, pp. 234–237, 2018.
- [11] R. Qi, C. Feng, Z. Liu, and N. Mrad, “Blockchain-Powered Internet of Things, E-Governance and E-Democracy,” in *E-Democracy for Smart Cities*, T. Vinod Kumar, Ed. Springer, may 2017, pp. 509–520.
- [12] P. Noizat, “Blockchain Electronic Vote,” in *Elsevier Handbook of Digital Currency*, may 2015, pp. 453–461.
- [13] K. Gai, Y. Wu, L. Zhu, L. Xu, and Y. Zhang, “Permissioned Blockchain and Edge Computing Empowered Privacy-Preserving Smart Grid Networks,” *IEEE Internet of Things J.*, vol. 6, no. 5, pp. 7992–8004, oct 2019.
- [14] F. Lombardi, L. Aniello, S. D. Angelis, A. Margheri, and V. Sassone, “A Blockchain-based Infrastructure for Reliable and Cost-effective IoT-aided Smart Grids,” in *IET Living in the Internet of Things: Cybersecurity of the IoT*, London, UK, mar 2018.
- [15] M. Mylrea and S. N. G. Gouriseti, “Blockchain for Smart Grid Resilience: Exchanging Distributed Energy at Speed, Scale and Security,” in *IEEE Resilience Week (RWS’17)*, Wilmington, DE, USA, sep 2017, pp. 18–23.
- [16] E. Mengelkamp, B. Notheisen, C. Beer, D. Dauer, and C. Weinhardt, “A blockchain-based smart grid: towards sustainable local energy markets,” *Springer Comput. Sci. Res. Dev.*, vol. 33, no. 1-2, pp. 207–214, aug 2017.
- [17] E. Munsing, J. Mather, and S. Moura, “Blockchains for Decentralized Optimization of Energy Resources in Microgrid Networks,” in *IEEE Conf. on Control Technol. and Appl. (CCTA’17)*, Mauna Lani, HI, USA, aug 2017, pp. 2164–2171.
- [18] A. Hahn, R. Singh, C.-C. Liu, and S. Chen, “Smart Contract-based Campus Demonstration of Decentralized Transactive Energy Auctions,” in *IEEE Power and Energy Soc. Innovative Smart Grid Technol. Conf. (ISGT’17)*, Arlington, VA, USA, apr 2017.
- [19] A. Laszka, A. Dubey, M. Walker, and D. Schmidt, “Providing privacy, safety, and security in IoT-based transactive energy systems using distributed ledgers,” in *Proc. of the 7th ACM Int. Conf. on the Internet of Things (IoT’17)*, Linz, Austria, 2017.
- [20] H. Yan, B.-B. Huang, and B.-W. Hong, “Distributed Energy Transaction Pattern and Block Chain Based Architecture Design,” *DEStech Trans. Environment, Energy Earth Sci.*, no. epee, feb 2018.
- [21] J. Xu, K. Xue, S. Li, H. Tian, J. Hong, P. Hong, and N. Yu, “Healthchain: A Blockchain-Based Privacy Preserving Scheme for Large-Scale Health Data,” *IEEE Internet of Things J.*, vol. 6, no. 5, pp. 8770–8781, oct 2019.
- [22] A. Dwivedi, G. Srivastava, S. Dhar, and R. Singh, “A Decentralized Privacy-Preserving Healthcare Blockchain for IoT,” *MDPI Sensors*, vol. 19, no. 2, p. 326, jan 2019.
- [23] M. Mettler, “Blockchain technology in healthcare: The revolution starts here,” in *Proc. of the 18th IEEE Int. Conf. on e-Health Networking, Applications and Services (Healthcom’18)*, Munich, Germany, sep 2016, pp. 1–3.
- [24] X. Yue, H. Wang, D. Jin, M. Li, and W. Jiang, “Healthcare Data Gateways: Found Healthcare Intelligence on Blockchain with Novel Privacy Risk Control,” *Springer J. of Medical Syst.*, vol. 40, no. 10, aug 2016.
- [25] L. Cocco, A. Pinna, and M. Marchesi, “Banking on Blockchain: Costs Savings Thanks to the Blockchain Technology,” *MDPI Future Internet*, vol. 9, no. 3, p. 25, jun 2017.



- [26] Y. Guo and C. Liang, "Blockchain application and outlook in the banking industry," *Springer Financial Innovation*, vol. 2, no. 1, dec 2016.
- [27] M. Shen, X. Tang, L. Zhu, X. Du, and M. Guizani, "Privacy-Preserving Support Vector Machine Training Over Blockchain-Based Encrypted IoT Data in Smart Cities," *IEEE Internet of Things J.*, vol. 6, no. 5, pp. 7702–7712, oct 2019.
- [28] S. Ibba, A. Pinna, M. Seu, and F. E. Pani, "CitySense: blockchain-oriented smart cities," in *Proc. of the XP2017 Scientific Workshops (XP'17)*, Cologne, Germany, 2017, pp. 1–5.
- [29] A. S. Patil, B. A. Tama, Y. Park, and K.-H. Rhee, "A Framework for Blockchain Based Secure Smart Green House Farming," *Springer, Advances in Computer Science and Ubiquitous Computing*, pp. 1162–1167, dec 2017.
- [30] K. Liu, W. Chen, Z. Zheng, Z. Li, and W. Liang, "A Novel Debt-Credit Mechanism for Blockchain-Based Data-Trading in Internet of Vehicles," *IEEE Internet of Things J.*, vol. 6, no. 5, pp. 9098–9111, oct 2019.
- [31] T. Jiang, H. Fang, and H. Wang, "Blockchain-Based Internet of Vehicles: Distributed Network Architecture and Performance Analysis," *IEEE Internet of Things J.*, vol. 6, no. 3, pp. 4640–4649, jun 2019.
- [32] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-Based Decentralized Trust Management in Vehicular Networks," *IEEE Internet of Things J.*, vol. 6, no. 2, pp. 1495–1505, apr 2019.
- [33] A. Patel, N. Shah, T. Limbasiya, and D. Das, "VehicleChain: Blockchain-based Vehicular Data Transmission Scheme for Smart City," in *Proc. of the IEEE Int. Conf. on Syst., Man, and Cybern. (SMC'19)*, Bari, Italy, oct 2019, pp. 661–667.
- [34] M. Cebe, E. Erdin, K. Akkaya, H. Aksu, and S. Uluagac, "Block4Forensic: An Integrated Lightweight Blockchain Framework for Forensics Applications of Connected Vehicles," *IEEE Commun. Magazine*, vol. 56, no. 10, pp. 50–57, oct 2018.
- [35] R. A. Michelin, A. Dorri, M. Steger, R. C. Lunardi, S. S. Kanhere, R. Jurdak, and A. F. Zorzo, "SpeedyChain: A framework for decoupling data from blockchain for smart cities," in *Proc. of the 15th EAI Int. Conf. on Mobile and Ubiquitous Syst.: Comput., Netw. and Services (MobiQui-tous'18)*, New York, NY, USA, nov 2018, pp. 145–154.
- [36] X. Zhang, R. Li, and B. Cui, "A security architecture of VANET based on blockchain and mobile edge computing," in *Proc. of the 1st IEEE Int. Conf. on Hot Information-Centric Networking (HotICN'18)*, Shenzhen, China, aug 2018, pp. 258–259.
- [37] L. Li, J. Liu, L. Cheng, S. Qiu, W. Wang, X. Zhang, and Z. Zhang, "CreditCoin: A Privacy-Preserving Blockchain-Based Incentive Announcement Network for Communications of Smart Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 7, pp. 2204–2220, jul 2018.
- [38] S. Rowan, M. Clear, M. Gerla, M. Huggard, and C. M. Goldrick, "Securing Vehicle to Vehicle Communications using Blockchain through Visible Light and Acoustic Side-Channels," *arXiv preprint arXiv:1704.02553*, apr 2017.
- [39] P. K. Sharma, S. Y. Moon, and J. H. Park, "Block-VN: A Distributed Blockchain Based Vehicular Network Architecture in Smart City," *KIPS Journal of Information Processing Systems*, vol. 13, no. 1, pp. 184–195, 2017.
- [40] M. Singh and S. Kim, "Intelligent Vehicle-Trust Point: Reward based Intelligent Vehicle Communication using Blockchain," *arXiv preprint arXiv:1707.07442*, 2017.
- [41] Z. Yang, K. Zheng, K. Yang, and V. C. M. Leung, "A blockchain-based reputation system for data credibility assessment in vehicular networks," in *IEEE 28th Annu. Int. Symp. on Pers., Indoor, and Mobile Radio Commun. (PIMRC'17)*, Montreal, QC, Canada, oct 2017, pp. 1–5.
- [42] Y. Yuan and F.-Y. Wang, "Towards blockchain-based intelligent transportation systems," in *IEEE 19th Int. Conf. on Intell. Transp. Syst. (ITSC'16)*, Rio de Janeiro, Brazil, nov 2016, pp. 2663–2668.
- [43] B. Leiding, P. Memarmoshrefi, and D. Hogrefe, "Self-managed and blockchain-based vehicular ad-hoc networks," in *Proc. of the ACM Int. Joint Conf. on Pervasive and Ubiquitous Computing: Adjunct (Ubi-Comp'16)*, Heidelberg, Germany, sep 2016, pp. 137–140.
- [44] A. Kapitonov, S. Lonshakov, A. Krupenkin, and I. Berman, "Blockchain-based protocol of autonomous business activity for multi-agent systems consisting of UAVs," in *IEEE Workshop on Research, Educ. and Dev. of Unmanned Aerial Syst. (RED-UAS'17)*, Linkoping, Sweden, oct 2017, pp. 84–89.
- [45] X. Liang, J. Zhao, S. Shetty, and D. Li, "Towards data assurance and resilience in IoT using blockchain," in *Proc. of the IEEE Military Commun. Conf. (MILCOM'17)*, Baltimore, MD, USA, oct 2017, pp. 261–266.
- [46] V. Sharma, I. You, and G. Kul, "Socializing Drones for Inter-Service Operability in Ultra-Dense Wireless Networks using Blockchain," in *Proc. of the 2017 ACM Int. Workshop on Managing Insider Security Threats (MIST'17)*, Dallas, TX, USA, oct 2017, pp. 81–84.
- [47] E. C. Ferrer, "The Blockchain: A New Framework for Robotic Swarm Systems," *Springer Advances in Intell. Syst. and Comput.*, pp. 1037–1058, oct 2018.
- [48] A. Kuzmin and E. Znak, "Blockchain-base structures for a secure and operate network of semi-autonomous Unmanned Aerial Vehicles," in *Proc. of the IEEE Int. Conf. on Service Operations and Logistics, and Informatics (SOLI'18)*, Singapore, jul 2018, pp. 32–37.
- [49] M. Wu, K. Wang, X. Cai, S. Guo, M. Guo, and C. Rong, "A Comprehensive Survey of Blockchain: From Theory to IoT Applications and Beyond," *IEEE Internet of Things J.*, vol. 6, no. 5, pp. 8114–8154, oct 2019.
- [50] W. Viriyasitavat, L. D. Xu, Z. Bi, and D. Hoonsopon, "Blockchain Technology for Applications in Internet of Things - Mapping From System Design Perspective," *IEEE Internet of Things J.*, vol. 6, no. 5, pp. 8155–8168, oct 2019.
- [51] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks," *IEEE Access*, vol. 7, pp. 22 328–22 370, jan 2019.
- [52] X. Xu, I. Weber, M. Staples, L. Zhu, J. Bosch, L. Bass, C. Pautasso, and P. Rimba, "A Taxonomy of Blockchain-Based Systems for Architecture Design," in *Proceedings of the IEEE Int. Conf. on Softw. Archit. (ICSA'17)*, Gothenburg, Sweden, apr 2017, pp. 243–252.
- [53] A. Panarello, N. Tapas, G. Merlino, F. Longo, and A. Puliafito, "Blockchain and IoT Integration: A Systematic Survey," *MDPI Sensors*, vol. 18, no. 8, p. 2575, aug 2018.
- [54] H. Wang, Z. Zheng, S. Xie, H. N. Dai, and X. Chen, "Blockchain challenges and opportunities: a survey," *Inderscience Int. J. Web and Grid Services*, vol. 14, no. 4, pp. 352–375, oct 2018.
- [55] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, oct 2017.
- [56] I.-C. Lin and T.-C. Liao, "A survey of blockchain security issues and challenges," *Int. J. of Network Security*, vol. 19, no. 5, pp. 653–659, sep 2017.
- [57] O. Novo, "Blockchain Meets IoT: An Architecture for Scalable Access Management in IoT," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1184–1195, apr 2018.
- [58] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, "IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices," *IEEE Internet of Things J.*, vol. 6, no. 5, pp. 8182–8201, oct 2019.
- [59] A. de Vries, "Bitcoin's Growing Energy Problem," *Elsevier Joule*, vol. 2, no. 5, pp. 801–805, may 2018.
- [60] S. Kim and G. C. Deka, *Advanced Applications of Blockchain Technology*. Springer, 2019.
- [61] M. Belotti, N. Bozic, G. Pujolle, and S. Secci, "A Vademecum on Blockchain Technologies: When, Which, and How," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3796–3838, jul 2019.
- [62] S. Ammous, "Blockchain Technology: What is it Good for?" *Elsevier SSRN Electronic J.*, aug 2016.
- [63] J. Garzik and J. C. Donnelly, "Blockchain 101: An Introduction to the Future," in *Handbook of Blockchain, Digital Finance, and Inclusion*. Elsevier, 2018, vol. 2, ch. 8, pp. 179–186.
- [64] Pérez-Marco. (2016) What is a blockchain? [Online]. Available: <https://webusers.imj-prg.fr/~ricardo.perez-marco/publications/talks/blockchain3.pdf> [Accessed: 2021/02/09].
- [65] D. Vujicic, D. Jagodic, and S. Randic, "Blockchain technology, bitcoin, and Ethereum: A brief overview," in *Proc. of the 17th IEEE Int. Symp. Infoteh-Jahorina (INFOTEH'18)*, East Sarajevo, Bosnia-Herzegovina, mar 2018, pp. 1–6.
- [66] Bitcoin Wiki. (2017, apr) Bitcoin Difficulty. <https://en.bitcoin.it/wiki/Difficulty>. [Online]. Available: <https://en.bitcoin.it/wiki/Difficulty> [Accessed: 2021/02/09].
- [67] C. Decker and R. Wattenhofer, "Information propagation in the Bitcoin network," in *IEEE Int. Conf. on Peer-to-Peer Computing (P2P'13)*, Trento, Italy, sep 2013.
- [68] N. T. Courtois, "On The Longest Chain Rule and Programmed Self-Destruction of Crypto Currencies," *arXiv preprint arXiv:1405.0534v11*, 2014.
- [69] E. Shi, "Analysis of Deterministic Longest-Chain Protocols," in *Proc. of the 32nd IEEE Comput. Security Found. Symp. (CSF'19)*, Hoboken, NJ, USA, jun 2019, pp. 122–12213.

- [70] Bitcoin Confirmation. [Online]. Available: <https://en.bitcoin.it/wiki/Confirmation> [Accessed: 2021/02/09].
- [71] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the Security and Performance of Proof of Work Blockchains," in *ACM SIGSAC Conf. on Computer and Communications Security (CCS'16)*, Vienna, Austria, oct 2016.
- [72] BitInfoCharts. Statistics about Block Generation Time for many popular Cryptocurrencies. <https://bitinfocharts.com/comparison/confirmationtime-btc-ltc-doge-bch-bsv-zec-xmr-dash-btg-rdd-blk.html#3m>. [Accessed: 2021/02/09].
- [73] P. Fairley, "Feeding the Blockchain Beast," *IEEE Spectrum*, vol. 54, pp. 36–59, oct 2017.
- [74] M. J. Fischer, N. A. Lynch, and M. S. Paterson, "Impossibility of Distributed Consensus with One Faulty Process," *J. of the ACM (JACM)*, vol. 32, no. 2, pp. 374–382, apr 1985.
- [75] S. Gilbert and N. Lynch, "Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-tolerant Web Services," *ACM SIGACT News*, vol. 33, no. 2, pp. 51–59, jun 2002.
- [76] D. Abadi, "Consistency Tradeoffs in Modern Distributed Database System Design: CAP is Only Part of the Story," *IEEE Computer*, vol. 45, no. 2, pp. 37–42, feb 2012.
- [77] R. Friedman and K. Birman, "Trading Consistency for Availability in Distributed Systems," *Cornell Computer Science Technical Reports*, apr 1996.
- [78] M. Al-Bassam, A. Sonnino, S. Bano, D. Hrycyszyn, and G. Danezis, "Chainspace: A sharded smart contracts platform," *arXiv preprint arXiv:1708.03778*, 2017.
- [79] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "OmniLedger: A Secure, Scale-Out, Decentralized Ledger via Sharding," in *Proc. of the IEEE Symp. on Security and Privacy (SP'18)*, San Francisco, CA, USA, may 2018, pp. 583–598.
- [80] F. Gräbe, N. Kannengießer, S. Lins, and A. Sunyaev, "Do Not Be Fooled: Toward a Holistic Comparison of Distributed Ledger Technology Designs," in *53rd Hawaii Int. Conf. on System Sciences (HICSS'20)*, Maui, HI, USA, jan 2020.
- [81] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 4, no. 3, pp. 382–401, jul 1982.
- [82] M. Castro, B. Liskov *et al.*, "Practical Byzantine Fault Tolerance," in *Proc. of the 3rd Symp. on Operating Syst. Design and Implementation*, New Orleans, LA, USA, feb 1999.
- [83] P. A. Bernstein, V. Hadzilacos, and N. Goodman, *Concurrency control and recovery in database systems*. Reading, Mass: Addison-Wesley Pub. Co, 1987.
- [84] L. Lamport, "Paxos made simple," *ACM Sigact News*, vol. 32, no. 4, pp. 18–25, 2001.
- [85] D. Ongaro and J. Ousterhout, "In Search of an Understandable Consensus Algorithm," in *Proc. of the USENIX Annu. Technical Conf. (USENIX-ATC'14)*, Philadelphia, PA, USA, jun 2014, pp. 305–319.
- [86] P.-L. Aublin, S. B. Mokhtar, and V. Quema, "RBFT: Redundant Byzantine Fault Tolerance," in *Proc. of the IEEE 33rd Int. Conf. on Distrib. Comput. Syst.*, Philadelphia, PA, USA, jul 2013, pp. 297–306.
- [87] M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, and I. Abraham, "Hot-Stuff: BFT Consensus in the Lens of Blockchain," *arXiv preprint arXiv:1803.05069*, 2018.
- [88] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S. W. Cocco, and J. Yellick, "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains," in *Proc. of the 13th ACM EuroSys Conf. (EuroSys'18)*, Porto, Portugal, apr 2018.
- [89] E. Buchman, J. Kwon, and Z. Milosevic, "The latest gossip on BFT consensus," *arXiv preprint arXiv:1807.04938*, 2018.
- [90] B. Chase and E. MacBrough, "Analysis of the XRP ledger consensus protocol," *arXiv preprint arXiv:1802.07242*, 2018.
- [91] J. Gray and L. Lamport, "Consensus on Transaction Commit," *ACM Trans. on Database Syst.*, vol. 31, no. 1, pp. 133–160, mar 2006.
- [92] L. S. Sabel and K. Marzullo, "Election vs. consensus in asynchronous systems," Cornell University, USA, Tech. Rep., 1995.
- [93] N. Santoro, "Election," in *Design and Analysis of Distributed Algorithms*. John Wiley & Sons, Inc., apr 2006, ch. 3, pp. 99–224.
- [94] M. Brooker. (2019) Leader election in distributed systems. [Online]. Available: <https://aws.amazon.com/builders-library/leader-election-in-distributed-systems> [Accessed: 2021/02/09].
- [95] M. Lokhava, G. Losa, D. Mazières, G. Hoare, N. Barry, E. Gafni, J. Jove, R. Malinowsky, and J. McCaleb, "Fast and secure global payments with stellar," in *Proc. of the 27th ACM Symp. on Operating Syst. Principles (SOSP'19)*, Huntsville, Ontario, Canada, oct 2019.
- [96] A. Montesor, "Gossip and Epidemic Protocols," *Wiley Encyclopedia of Elect. and Electron. Eng.*, pp. 1–15, aug 2017.
- [97] L. Baird, "The Swirls Hashgraph Consensus Algorithm: Fair, Fast, Byzantine Fault Tolerance," Swirls, Tech. Rep. SWIRLDS-TR-2016-01, may 2016. [Online]. Available: <https://www.swirls.com/downloads/SWIRLDS-TR-2016-01.pdf>
- [98] M. Ahamad, G. Neiger, J. E. Burns, P. Kohli, and P. W. Hutto, "Causal memory: definitions, implementation, and programming," *Springer Distributed Computing*, vol. 9, no. 1, pp. 37–49, mar 1995.
- [99] L. Lamport, "How to Make a Multiprocessor Computer That Correctly Executes Multiprocess Programs," *IEEE Trans. Comput.*, vol. C-28, no. 9, p. 690–691, sep 1979.
- [100] M. M. Elbushra and J. Lindström, "Causal Consistent Databases," *Ron-Pub Open J. of Databases (OJDB)*, vol. 2, no. 1, pp. 17–35, 2015.
- [101] L. M. Bach, B. Mihaljevic, and M. Zagar, "Comparative analysis of blockchain consensus algorithms," in *Proc. of the 41st IEEE Int. Conf. on Inform. and Commun. Technol., Electron. and Microelectron. (MIPRO'18)*, Opatija, Croatia, may 2018, pp. 1545–1550.
- [102] L. Chen, L. Xu, N. Shah, Z. Gao, Y. Lu, and W. Shi, "On Security Analysis of Proof-of-Elapsed-Time (PoET)," in *Proc. of the 19th Springer Int. Symp. on Stabilization, Safety, and Security of Distrib. Syst. (SSS'17)*, P. Spirakis and P. Tsigas, Eds., Boston, MA, USA, oct 2017, pp. 282–297.
- [103] L. Ghio, L. Maccari, and R. Lo Cigno, "Proof of Networking: Can Blockchains Boost the Next Generation of Distributed Networks?" in *Proc. of the 14th IEEE Conf. on Wireless On-demand Netw. Syst. and Services (WONS'18)*, Isola, France, feb 2018, pp. 29–32.
- [104] T. Larsson and R. Thorsén. (2018, jun) Cryptocurrency performance analysis of Burstcoin mining. [Online]. Available: <https://tinyurl.com/y78uv43a> [Accessed: 2021/02/09].
- [105] V. Buterin and V. Griffith, "Casper the friendly finality gadget," *arXiv preprint arXiv:1710.09437*, 2017.
- [106] J. Poon and T. Dryja. (2016) The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. [Online]. Available: <https://lightning.network/lightning-network-paper.pdf> [Accessed: 2021/02/09].
- [107] J. Tremback, J. Kilpatrick, D. Simpier, and B. Wang. (2017) Althea whitepaper. [Online]. Available: <https://althea.net/whitepaper> [Accessed: 2021/02/09].
- [108] Monero homepage. [Online]. Available: <https://web.getmonero.org> [Accessed: 2021/02/09].
- [109] S. Adve and K. Gharachorloo, "Shared memory consistency models: a tutorial," *Computer*, vol. 29, no. 12, pp. 66–76, dec 1996.
- [110] Wikipedia. List of cryptocurrencies. [Online]. Available: [https://en.wikipedia.org/wiki/List\\_of\\_cryptocurrencies](https://en.wikipedia.org/wiki/List_of_cryptocurrencies) [Accessed: 2021/02/09].
- [111] F. Armknecht, J.-M. Bohli, G. O. Karame, and W. Li, "Sharding PoW-based Blockchains via Proofs of Knowledge," *IACR Cryptol. ePrint Arch.*, vol. 2017, p. 1067, 2017.
- [112] Cryptocurrency Prices by Market Cap. [Online]. Available: <https://coinmarketcap.com> [Accessed: 2021/02/09].
- [113] E. Politou, F. Casino, E. Alepis, and C. Patsakis, "Blockchain Mutability: Challenges and Proposed Solutions," *IEEE Trans. Emerg. Topics Comput.*, pp. 1–1, oct 2019.
- [114] D. G. Bobrow, "A Note on Hash Linking," *Communications of the ACM*, vol. 18, no. 7, p. 413–415, jul 1975.
- [115] C. Halatsis and G. Philokyprou, "Pseudochaining in Hash Tables," *Communications of the ACM*, vol. 21, no. 7, pp. 554–557, jul 1978.
- [116] M. Iansiti and K. R. Lakhani, "The Truth About Blockchain," *Harvard Business Review*, pp. 118–127, jan 2017.
- [117] D. Floyd. (2019, jun) Banks Claim They're Building Blockchains. They're Not. [Online]. Available: <https://investopedia.com/news/banks-building-blockchains-distributed-ledger-permission/> [Accessed: 2021/02/09].
- [118] J. Göbel and A. E. Krzesinski, "Increased block size and Bitcoin blockchain dynamics," in *Proc. of 27th IEEE Int. Telecommunication Networks and Applications Conf. (ITNAC'17)*, Melbourne, Australia, nov 2017.
- [119] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. Gün Sirer, D. Song, and R. Wattenhofer, "On Scaling Decentralized Blockchains," in *Springer Int. Conf. on Financial Cryptography and Data Security (FC'16)*, Christ Church, Barbados, feb 2016, pp. 106–125.
- [120] M. Vukolić, "The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication," in *Springer Int. Workshop on Open Problems in*

- Network Security (iNetSec'15)*, Zurich, Switzerland, oct 2016, pp. 112–125.
- [121] Blockchain.com. Bitcoin Hashrate Distribution among Pools. <https://www.blockchain.com/charts/pools>. [Accessed: 2021/02/09].
- [122] A. E. Gencer, S. Basu, I. Eyal, R. Van Renesse, and E. G. Sirer, “Decentralization in Bitcoin and Ethereum Networks,” in *Springer Int. Conf. on Financial Cryptography and Data Security (FC'18)*, Nieuwpoort, Curaçao, Netherlands, mar 2018, pp. 439–457.
- [123] A. Gervais, G. O. Karame, V. Capkun, and S. Capkun, “Is Bitcoin a Decentralized Currency?” *IEEE Security & Privacy*, vol. 12, no. 3, pp. 54–60, may 2014.
- [124] I. Eyal and E. G. Sirer, “Majority is not enough: Bitcoin mining is vulnerable,” in *18th Springer Int. Conf. on Financial Cryptography and Data Security (FC'14)*, Christ Church, Barbados, mar 2014, pp. 436–454.
- [125] M. Carlsten, H. Kalodner, S. M. Weinberg, and A. Narayanan, “On the Instability of Bitcoin Without the Block Reward,” in *ACM SIGSAC Conf. on Computer and Communications Security (CCS'16)*, Vienna, Austria, oct 2016.
- [126] G. Fanti, L. Kogan, S. Oh, K. Ruan, P. Viswanath, and G. Wang, “Compounding of Wealth in Proof-of-Stake Cryptocurrencies,” in *23rd Springer Int. Conf. on Financial Cryptography and Data Security (FC'19)*, Frigate Bay, St. Kitts and Nevis, 2019, pp. 42–61.
- [127] C. Catalini, R. Jagadeesan, and S. D. Kominers, “Market Design for a Blockchain-Based Financial System,” *SSRN Electronic J.*, 2019.
- [128] W. K. Härdle, C. R. Harvey, and R. C. G. Reule, “Understanding Cryptocurrencies,” *Oxford University Press, J. of Financial Econometrics*, vol. 18, no. 2, pp. 181–208, 02 2020.
- [129] G. Fanti, L. Kogan, and P. Viswanath, “Economics of Proof-of-Stake Payment Systems,” 2020, PRELIMINARY AND INCOMPLETE, Latest draft: March 2020. [Online]. Available: [http://fetch.econ.cam.ac.uk/papers/main\\_3\\_25\\_2020-Kogan.pdf](http://fetch.econ.cam.ac.uk/papers/main_3_25_2020-Kogan.pdf)
- [130] E. Deirmentzoglou, G. Papakriakopoulos, and C. Patsakis, “A Survey on Long-Range Attacks for Proof of Stake Protocols,” *IEEE Access*, vol. 7, pp. 28 712–28 725, 2019.
- [131] P. Gazi, A. Kiayias, and A. Russell, “Stake-Bleeding Attacks on Proof-of-Stake Blockchains,” in *IEEE Crypto Valley Conf. on Blockchain Technology (CVCBT'18)*, Zug, Switzerland, jun 2018.
- [132] J. Li, N. Li, J. Peng, H. Cui, and Z. Wu, “Energy consumption of cryptocurrency mining: A study of electricity consumption in mining cryptocurrencies,” *Elsevier Energy*, vol. 168, pp. 160–168, feb 2019.
- [133] M. E. Peck, “Blockchain world - Do you need a blockchain? This chart will tell you if the technology can solve your problem,” *IEEE Spectrum*, vol. 54, no. 10, pp. 38–60, oct 2017.
- [134] K. Wüst and A. Gervais, “Do you need a Blockchain?” in *Proc. of the IEEE Crypto Valley Conf. on Blockchain Technol. (CVCBT'18)*, Zug, Switzerland, jun 2018, pp. 45–54.
- [135] S. Meunier. When do you need blockchain? decision models. [Online]. Available: <https://medium.com/@sbmeunier/when-do-you-need-blockchain-decision-models-a5c40e7c9ba1> [Accessed: 2021/02/09].
- [136] S. Heiberg, I. Kubjas, J. Siim, and J. Willemson, “On Trade-offs of Applying Block Chains for Electronic Voting Bulletin Boards,” in *3rd Int. Joint Conf. on Electronic Voting (E-Vote-ID'18)*, Lochau/Bregenz, Austria, oct 2018, pp. 259–276.
- [137] M. M. H. Onik and M. H. Miraz, “Performance Analytical Comparison of Blockchain-as-a-Service (BaaS) Platforms,” in *Proc. of the 2nd Springer Int. Conf. on Emerging Technol. in Comput. (iCETiC'19)*, London, UK, aug 2019, pp. 3–18.
- [138] C. Gorenflo, S. Lee, L. Golab, and S. Keshav, “FastFabric: Scaling Hyperledger Fabric to 20,000 Transactions per Second,” in *Proc. of the IEEE Int. Conf. on Blockchain and Cryptocurrency (ICBC'19)*, Seoul, South Korea, may 2019.
- [139] Hyperledger and its projects. [Online]. Available: <https://hyperledger.org/learn> [Accessed: 2021/02/09].
- [140] Hyperledger Fabric Channels. [Online]. Available: <https://github.com/hyperledger/fabric/blob/master/docs/source/channels.rst> [Accessed: 2021/02/09].
- [141] Hyperledger Fabric Private data. [Online]. Available: <https://hyperledger-fabric.readthedocs.io/en/latest/private-data/private-data.html> [Accessed: 2021/02/09].
- [142] S. Popejoy. (2019, jul) Why ibm’s blockchain isn’t a real blockchain. [Online]. Available: <https://cointelegraph.com/news/why-ibms-blockchain-isnt-a-real-blockchain> [Accessed: 2021/02/09].
- [143] Hyperledger Sawtooth Homepage. [Online]. Available: <https://sawtooth.hyperledger.org> [Accessed: 2021/02/09].
- [144] PoET 1.0 Specification. [Online]. Available: <https://sawtooth.hyperledger.org/docs/core/releases/1.0/architecture/poet.html> [Accessed: 2021/02/09].
- [145] Known Attacks to Software Guard Extensions. [Online]. Available: [https://en.wikipedia.org/wiki/Software\\_Guard\\_Extensions#Attacks](https://en.wikipedia.org/wiki/Software_Guard_Extensions#Attacks) [Accessed: 2021/02/09].
- [146] M. Frustaci, P. Pace, G. Aloï, and G. Fortino, “Evaluating Critical Security Issues of the IoT World: Present and Future Challenges,” *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2483–2495, aug 2018.
- [147] B. Lee and J.-H. Lee, “Blockchain-based secure firmware update for embedded devices in an Internet of Things environment,” *Springer J. of Supercomput.*, vol. 73, no. 3, pp. 1152–1167, sep 2016.
- [148] A. Mostefaoui, E. Mourgaya, and M. Raynal, “An introduction to oracles for asynchronous distributed systems,” *Elsevier Future Generation Comput. Syst.*, vol. 18, no. 6, pp. 757–767, may 2002.
- [149] J. Adler, R. Berryhill, A. Veneris, Z. Poulos, N. Veira, and A. Kastania, “Astraea: A Decentralized Blockchain Oracle,” in *IEEE Int. Conf. on Internet of Things (iThings) and IEEE Green Comput. and Commun. (GreenCom) and IEEE Cyber, Physical and Social Comput. (CPSCom) and IEEE Smart Data (SmartData)*, Halifax, NS, Canada, jul 2018, pp. 1145–1152.
- [150] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, “Blockchain for IoT security and privacy: The case study of a smart home,” in *Proc. of the IEEE Int. Conf. on Pervasive Comput. and Commun. Workshops (PerCom'17)*, Kona, HI, USA, mar 2017, pp. 618–623.
- [151] A. Ouaddah, A. A. Elkalam, and A. A. Ouahman, “Towards a Novel Privacy-Preserving Access Control Model Based on Blockchain Technology in IoT,” in *Proc. of the Springer Europe and MENA Cooperation Advances in Inf. and Commun. Technol. (EMENA-TSSL'16)*, Saidia, Oujda, Morocco, sep 2017, pp. 523–533.
- [152] Y. Rahulamathavan, R. C.-W. Phan, M. Rajarajan, S. Misra, and A. Kondozi, “Privacy-preserving blockchain based IoT ecosystem using attribute-based encryption,” in *Proc. of the IEEE Int. Conf. on Advanced Netw. and Telecommun. Syst. (ANTS'17)*, Bhubaneswar, India, dec 2017, pp. 1–6.
- [153] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, “LSB: A Lightweight Scalable Blockchain for IoT Security and Privacy,” *arXiv preprint arXiv:1712.02969*, 2017.
- [154] A. Dorri, S. S. Kanhere, and R. Jurdak, “Towards an optimized blockchain for IoT,” in *Proc. of the 2nd IEEE/ACM Int. Conf. on Internet-of-Things Des. and Implement. (IoTDP'17)*, Pittsburgh, PA, USA, apr 2017, pp. 173–178.
- [155] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak, “BlockChain: A Distributed Solution to Automotive Security and Privacy,” *IEEE Commun. Magazine*, vol. 55, no. 12, pp. 119–125, dec 2017.
- [156] G. Ateniese, B. Magri, D. Venturi, and E. Andrade, “Redactable Blockchain – or – Rewriting History in Bitcoin and Friends,” in *Proc. of the IEEE Eur. Symp. on Security and Privacy (EuroSP'17)*, Paris, France, apr 2017, pp. 111–126.
- [157] M. Berberich and M. Steiner, “Practitioner’s Corner - Blockchain Technology and the GDPR - How to Reconcile Privacy and Distributed Ledgers?” *Lexxon Eur. Data Protection Law Rev.*, vol. 2, no. 3, pp. 422–426, mar 2016.
- [158] G. Gabison, “Policy Considerations for the Blockchain Technology Public and Private Applications,” *SMU Sci. and Technol. Law Rev.*, vol. 19, no. 3, pp. 327–350, 2016.
- [159] P.-Y. Chang, M.-S. Hwang, and C.-C. Yang, “A Blockchain-Based Traceable Certification System,” in *Proc. of the Springer Int. Conf. on Security with Intell. Comput. and Big-data Services (SICBS'18)*, Cham, mar 2018, pp. 363–369.
- [160] W. Gräther, S. Kolvenbach, R. Ruland, J. Schütte, C. Torres, and F. Wendland, “Blockchain for Education: Lifelong Learning Passport,” in *Proc. of 1st ERCIM Blockchain Workshop 2018*, Amsterdam, Netherlands, may 2018.
- [161] M. D. Fowler, “Linking the Public Benefit to the Corporation: Blockchain as a Solution for Certification in an Age of Do-Good Business,” *Vanderbilt J. Entertainment and Technol. Law*, vol. 20, no. 3, p. 881, mar 2018.
- [162] R. Burstall and B. Clark, “Blockchain, IP and the Fashion Industry,” *Managing Intell. Prop.*, vol. 266, p. 9, apr 2017.
- [163] J.-C. Cheng, N.-Y. Lee, C. Chi, and Y.-H. Chen, “Blockchain and smart contract for digital certificate,” in *Proc. of the IEEE Int. Conf. on Applied Syst. Invention (ICASI'18)*, Chiba, Japan, apr 2018, pp. 1046–1051.
- [164] W. Nowiński and M. Kozma, “How Can Blockchain Technology Disrupt the Existing Business Models?” *CEEOL Entrepreneurial Business and Economics Review*, vol. 5, no. 3, pp. 173–188, jun 2017.

- [165] J. F. Galvez, J. Mejuto, and J. Simal-Gandara, "Future challenges on the use of blockchain for food traceability analysis," *Elsevier TrAC Trends in Analytical Chemistry*, vol. 107, pp. 222–232, oct 2018.
- [166] A. Bahga and V. K. Madiseti, "Blockchain Platform for Industrial Internet of Things," *SCIRP J. of Softw. Eng. and Appl.*, vol. 09, no. 10, pp. 533–546, oct 2016.
- [167] V. A. J. Boehm, J. Kim, and J. W.-K. Hong, "Holistic Tracking of Products on the Blockchain Using NFC and Verified Users," in *Proc. of the Springer Int. Workshop on Inf. Security Appl. (WISA'17)*, Jeju Island, Korea, aug 2017, pp. 184–195.
- [168] E. Perez. (2019, jul) Top-5 Famous Crypto Tokens That Seem "Dead": Analysis. [Online]. Available: <https://cointelegraph.com/news/top-5-crypto-tokens-pronounced-dead-nem-and-bcc-head-the-list> [Accessed: 2021/02/09].
- [169] M. Bartoletti, S. Carta, T. Cimoli, and R. Saia, "Dissecting Ponzi schemes on Ethereum: Identification, analysis, and impact," *Elsevier Future Generation Comput. Syst.*, vol. 102, pp. 259–277, jan 2020.
- [170] S. Dziembowski, S. Faust, and K. Hostáková, "General State Channel Networks," in *Proc. of the 2018 ACM SIGSAC Conf. on Computer and Comm. Security (CCS'18)*, jan 2018, p. 949–966.
- [171] P. McCorry, M. Möser, S. F. Shahandasti, and F. Hao, "Towards Bitcoin Payment Networks," in *Proc. of the Springer Australasian Conf. on Information Security and Privacy*, Christchurch, New Zealand, jun 2016, pp. 57–76.
- [172] A. Ensor, S. Schefer-Wenzl, and I. Miladinovic, "Blockchains for IoT Payments: A Survey," in *Proc. of the IEEE Globecom Workshops (GC Wkshps)*, Abu Dhabi, United Arab Emirates, dec 2018, pp. 1–6.
- [173] A. Miller, I. Bentov, S. Bakshi, R. Kumaresan, and P. McCorry, "Sprites and State Channels: Payment Networks that Go Faster Than Lightning," in *Proc. of the Int. Conf. on Financial Cryptography and Data Security (FC'19)*, Saint Kitts and Nevis, sep 2019, pp. 508–526.
- [174] C. Burchert, C. Decker, and R. Wattenhofer, "Scalable funding of Bitcoin micropayment channel networks," *Royal Society Open Science*, vol. 5, 180089, pp. 1–15, aug 2018.
- [175] V. Sivaraman, S. B. Venkatakrisnan, K. Ruan, P. Negi, L. Yang, R. Mittal, G. Fanti, and M. Alizadeh, "High Throughput Cryptocurrency Routing in Payment Channel Networks," in *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*, Santa Clara, CA, feb 2020, pp. 777–796.
- [176] L. Maccari and R. Lo Cigno, "Improving Routing Convergence With Centrality: Theory and Implementation of Pop-Routing," *IEEE/ACM Trans. on Networking*, vol. 26, no. 5, pp. 2216–2229, oct 2018.
- [177] L. Maccari, L. Ghio, A. Guerrieri, A. Montresor, and R. Lo Cigno, "Exact Distributed Load Centrality Computation: Algorithms, Convergence, and Applications to Distance Vector Routing," *IEEE Trans. on Parallel and Distrib. Syst.*, vol. 31, no. 7, pp. 1693–1706, jul 2020.
- [178] R. Pickhardt and M. Nowostawski, "Imbalance measure and proactive channel rebalancing algorithm for the Lightning Network," *arXiv preprint arXiv:1912.09555*, 2019.
- [179] S. Tikhomirov, R. Pickhardt, A. Biryukov, and M. Nowostawski, "Probing Channel Balances in the Lightning Network," *arXiv preprint arXiv:2004.00333*, 2020.
- [180] L. Eckey, S. Faust, K. Hostáková, and S. Roos, "Splitting Payments Locally While Routing Interdimensionally," *Cryptology ePrint Archive*, may 2020.