A Blockchain Definition to Clarify its Role for the Internet of Things

Lorenzo Ghiro*, Francesco Restuccia†, Salvatore D'Oro†,
Stefano Basagni†, Tommaso Melodia†, Leonardo Maccari‡, Renato Lo Cigno§
*University of Trento, Italy, lorenzo.ghiro@unitn.it
†Northeastern University, USA, {f.restuccia, s.doro, s.basagni, t.melodia}@northeastern.edu
‡University of Venice, Italy, leonardo.maccari@unive.it
§University of Brescia, Italy, renato.locigno@unibs.it

***Abstract*—The term *blockchain* is used for disparate projects, ranging from cryptocurrencies to applications for the Internet of Things (IoT). The concept of blockchain appears therefore blurred, as the same technology cannot empower applications with extremely different requirements, levels of security and performance. This position paper elaborates on the theory of distributed systems to advance a clear definition of blockchain allowing us to clarify its possible role in the IoT. The definition binds together three elements that, as a whole, delineate those unique features that distinguish the blockchain from other distributed ledger technologies: *immutability*, *transparency* and *anonymity*. We note that immutability—which is imperative for securing blockchains—imposes remarkable resource consumption. Moreover, while transparency demands no confidentiality, anonymity enhances privacy but prevents user identification. As such, we raise the concern that these blockchain features clash with the requirements of most IoT applications where devices are power-constrained, data needs to be kept confidential, and users to be clearly identifiable. We consequently downplay the role of the blockchain for the IoT: this can act as a ledger *external* to the IoT architecture, invoked as seldom as possible and only to record the aggregate results of myriads of local (IoT) transactions that are most of the time performed off-chain to meet performance and scalability requirements.**

I. Introduction

The *blockchain* came into the limelight with the advent of the Bitcoin, which is the most successful blockchain application to date, hitting a US$1 trillion market capitalization in February 2021—a new record. Some features observed in Bitcoin, i.e., decentralization, resistance to powerful cyberattacks and preservation of user privacy, raised the enthusiasm of many communities, leading to an explosion of disparate proposals for using the blockchain in many different applications comprising Supply Chain Management [1], E-Voting [2], Smart Grid [3], Healthcare [4], Banking [5], Smart Cities [6], [7], and even Vehicular and Aerial Networks [8], [9], to name a few. Surveys focusing on the applications of the blockchain for the Internet of Things (IoT), for instance, already abound [10], [11].

This vast application range makes the blockchain look like an almost *universal* technology. We note however that the original Bitcoin blockchain supports less than 10 Transactions per Second (TPS) and consumes as much power as Ireland [12]: it is therefore unclear how a similar blockchain will ever be so versatile to support all of the documented applications, especially the IoT ones involving millions of TPS and tight power constraints. Indeed, moving to application domains different from cryptocurrencies, the characteristics of the original blockchain have been completely transformed, leading to "mutated blockchains" that are possible source of misunderstanding and confusion. On the one hand, we still have the *permissionless* blockchains like the original Bitcoin, celebrated for their Proof of Work (PoW)-based cryptographical security, their decentralization and strict privacy defense through anonymity. On the other hand, many more recent "blockchains" are *permissioned*, require user identities, and their internal security does not depend on some hard cryptographical problem such as the PoW. The single term blockchain appears therefore overloaded, resulting ambiguous, as it is used to indicate ledger technologies that address security, performance, and decentralization in completely different ways.

This position paper analyzes the multiple technologies proffered under the term *blockchain* from a distributed systems perspective, and proposes a clear definition of blockchain that allows arguing its role in the IoT. Our definition identifies three elements that, only when combined together, give to blockchains their specific features of openness, decentralization, security, and privacy: i) a Strong Distributed Consensus Protocol, which makes the blockchain immutable and frees them from centralized trusted authorities; ii) a Full & Public History of Transactions, which allows their distributed and completely transparent validation; and iii) to be Open to Anonymous Users, allowing complete users privacy.

To convey our crucial blockchain definition and the arguments on the blockchain role in the IoT, this paper is structured as follows. Sect. II critically analyses the genesis of the blockchain, stressing on the technical reasons motivating the design of a blockchain like Bitcoin. Sect. III discusses the trade-offs inherent to the design of consensus protocols, that are key for determining the properties of any shared ledger, either blockchain-based or not. Sect. II and III are the basis for the formulation of our blockchain definition, reported in the main Sect. IV. In light of the provided definition, in Sect. V we outline the marginal role we envision for the blockchain in the IoT.

An extended version of this paper, available on arXiv [13], includes a more complete background with a review of consensus protocols as well as a discussion about what we call *"The blockchain pitfalls,"* i.e., those common abuses of the blockchain in applications whose requirements contrast with the features of the same blockchain. Moreover it contains an expanded section on possible applications of the blockchain to the IoT without violating its constituent characteristics, omitted hire for space constraints.

II. Blockchain Fundamentals

Fig. 1 illustrates the general life-cycle of a transaction in a blockchain system. A user that issues a new transaction announces it in the Peer-to-Peer (P2P) network and waits for the correctness check performed by validator nodes. These nodes run a consensus protocol to determine if the issuer owns the resources it is spending or not. A transaction that is considered valid is grouped with others to form a new block of transactions, and this block is later registered in the ledger by appending it to the blockchain. At the end, the success of the transaction is notified to the users.

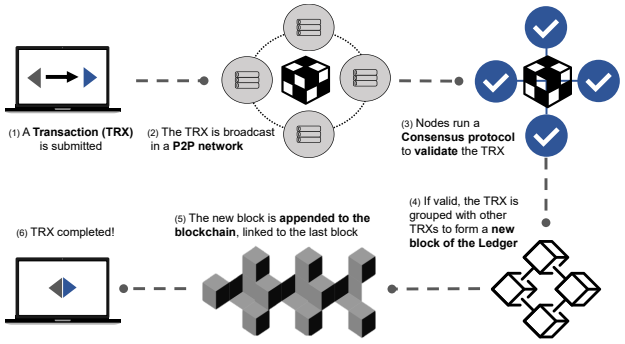

Figure 1. Processing of a transaction before storage in the blockchain.

At first glance, the blockchain may look as a plain data structure used to record transactions; however, from a broader perspective, a blockchain is a distributed system that includes:

itemsep=0em, leftmargin=0.5cm

- A *P2P network* made of all those nodes that either read or cooperatively write transactions in the blockchain, and
- 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 can thus be seen as a possible implementation of a Shared Ledger, which can be either *public* or *private* depending on who is allowed to append new transactions to the blockchain.

A. Public vs. Private 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. Users are thus not accountable, and security must be enforced through a technical solution. For example, in permissionless blockchains such as Bitcoin and Ethereum the anonymous proposer of a new block must provide the solution to a very hard crypto-problem, the so-called Proof of Work (PoW). On the one hand, the PoW proves the honest commitment of the proposer, but on the other hand it hampers performance and gobbles (computational) resources.

Private (Permissioned) ledgers arose as an attempt to improve performance and to have more control on users. A shared and mutual level of trust is 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. The resulting model allows blockchain managers to replace the resource-hungry consensus protocols of permissionless blockchains with more traditional, efficient, and faster ones.

B. The Need of the Transactions History

Validators need the history of transactions to determine who owns resources and how many. However, building this history in a distributed system is complicated by the double spending problem. This problem arises from transactions that spend the same resources "twice", but are received by distinct validators in diverse orders because of different propagation delays. Validators need to run a consensus protocol for sorting transactions into a unique chronological order, fundamental to determine which among two conflicting transactions should be considered first (valid) and which other second (rejected). The blockchain captures this chronological history of transactions, necessary for validation purposes, by grouping transactions in timestamped blocks. Unfortunately, the transactions history may not be enough. Indeed, a malicious user can alter the content of a block to repudiate an unwanted transaction, falsifying this way the validation procedure. To fend off falsification attacks, a blockchain must be *Tamper-proof* and *Immutable*. The tamper-proof property is achieved by a clever embedding of Cryptographic Hash Functions (CHFs) into the blockchain data structure, while the PoW makes the blockchain immutable as well.

C. The Proof of Work (PoW)

The Bitcoin protocol dictates that a block is valid only if the the application of the mandated CHF (i.e., double-SHA256) to the block content produces a digest smaller than a given target. The smaller the target the higher will be the required number of leading zeros in the most significant digits of the digest, a number we call Z. Notice that the double-SHA256 produces digests of 256 bits and that a miner can include an arbitrary piece of information in the block, the *nonce*, to influence the result of the CHF. The probability for a random nonce to lead to a valid digest can be approximately computed as a function of Z, resulting to be equal to $P(n) \approx 2^{-Z}$. The Z number is an indicator of the mining difficulty and can be adjusted to tune the generation rate of valid blocks. Finding a valid nonce is the *proof of work* (PoW), i.e., the proof of the effort in terms of computing power and energy spent to find such nonce.

D. Block Generation/Propagation and Forks

Bitcoin targets an average Block Generation Interval (B_{GI}) of 10 minutes. Fig. 2 reveals how the growth of the network computing power over time results in the need for dynamically adjusting Z to keep the desired target B_{GI}. The B_{GI} must be kept high to avoid the production of two simultaneous blocks. Two blocks are considered "simultaneous" if the second one is generated within the average block propagation time (B_P) of the first one. In general, the B_P of a block of a few megabytes (MB) in any P2P overlay is in the order of seconds to a maximum of a few tens of seconds [14]. Simultaneous blocks divide the nodes of the Bitcoin network into two parts that will append the two different blocks to the blockchain forming two branches: a "*fork*." The existence of a fork means that there is no consensus on the order of blocks (i.e., transactions), therefore the system is again exposed to

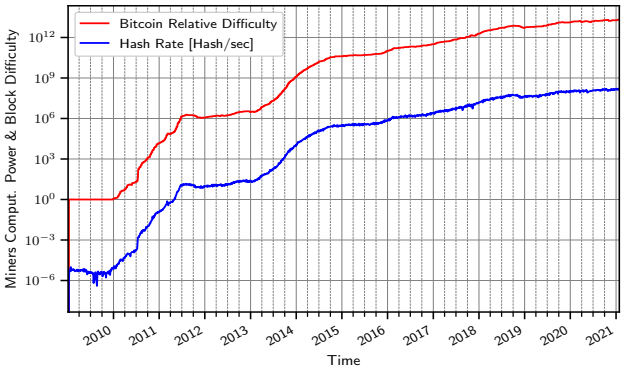

Figure 2. Evolution of the Bitcoin network computing power, measured in hash per second. Over time the block difficulty has been adjusted to keep a constant Block Generation Interval B_{GI}. Statistics are taken from https://www.blockchain.com/

double spending attacks. Usually forks are transient and are cleared as soon as another block is presented, making one branch longer than the other. In general, validators will prefer and give their consensus to this longer branch because of the so-called "*longest-chain rule*" [15]. According to this rule, only the blocks of the longest chain are valid, while the blocks on "stale" branches are not, meaning also that validators that want to gain the Bitcoin reward by submitting valid blocks are encouraged to dedicate their computational power only to the growth of the main chain.

One intrinsic limit of a blockchain is the presence of a propagation time necessary to distribute the knowledge of a newly created block in the peer-to-peer network of miners. Assume that all miners are competing to mine block b_i, and at time T_i a miner generates the block. After B_P seconds all the miners know that they should stop mining b_i and start mining b_{i+1}, but in the time interval $[T_i, T_i + B_P]$ miners are using resources to mine a block that will most likely not enter the blockchain. As an example B_P to reach 90% of the Bitcoin miners is in the order of tens of seconds[1]. One goal of a blockchain is thus to reduce the relative amount of wasted resources, that depends on the ratio between B_P and the average block creation time B_{GI}. This is referred to as the blockchain overhead [16]:

$$Oh \propto \frac{B_P}{B_{GI}} \tag{1}$$

A large B_{GI} is essential to maintain a low overhead, and prevents the deployment of a fast blockchain.

E. PoW and Immutability

Consider a malicious user that wants to cancel an unwanted transaction from, let's say, block i. This alteration would invalidate block i digest and, due to block chaining, also all the following ones, so that all validators would immediately notice the manipulation and refuse the tampered branch.

Another attack strategy exploits the longest chain rule and consists in generating a longer branch of blocks that replaces the previous one starting from block $i-1$. This attack is usually referred to as the 51% attack, as it becomes likely when some miner owns the majority of the mining resources [17], so it is extremely important that the mining power is not monopolized by one or a coalition of miners.

[1]See Decker et al. [14] and recent statistics from the Bitcoin blockchain http://bitcoinstats.com/network/propagation/.

III. The Limits of Consensus Protocols

Sect. II posed the distributed consensus problem on the order of transactions and explained how the PoW solves it. It turned out that the PoW must be necessarily power-hungry to make the cost of chain replacement attacks prohibitive. In general, the PoW advantages are many: it is secure, fully distributed, and user-agnostic. Ultimately, the PoW i) protects the user privacy and ii) frees users from trusted authorities. The popularity of blockchains, above all with cryptocurrencies, is most probably grounded in these two key aspects.

It can be observed that consensus protocols are a crucial component of 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. However, a question arises: Is it possible to design a consensus protocol that preserves the PoW advantages but, at the same time, avoids its drawbacks to meet the typical requirements of IoT applications? In the rest of this section, we briefly review the theory of distributed consensus protocols, reviving those theorems that limit the design of consensus protocols for blockchains in general, representing crucial bounds especially for IoT applications. We omit a full review of consensus protocols, which is available in [13].

The CAP[2] theorem [18] is a pillar of the theory of distributed systems and states that, whenever a system gets $\mathcal{P}$artitioned, then only two options are available: i) grant $\mathcal{C}$onsistency by safely blocking the system to fix the failures; or ii) keep processing transactions favoring $\mathcal{A}$vailability, with the risk that the two conflicting (double-spending) transactions could be recorded, one per partition. The CAP theorem may be considered only mildly relevant since it is valid only for ill-behaving systems, while in practice a system is built to work properly for most of its lifetime. However, it is the anteroom for the definition of two trade-offs of tremendous practical importance.

The first trade-off is known as PACELC [19], which advances the CAP theorem (shuffling the acronym) adding: *Else Latency or Consistency*. The PACELC theorem focuses on the trade-off between $\mathcal{L}$atency and $\mathcal{C}$onsistency arising from the propagation delays inescapable for any distributed system, thus valid also for not partitioned systems. In Sect. II, we have analyzed the Bitcoin design choices to calibrate this trade-off, showing how Bitcoin clearly favors consistency (security) over availability by setting the quite high B_{GI} of 10 minutes that, together with the 6 confirmations rule,[3] introduces a large delay to record transactions in the ledger.

The second trade-off is known as the *"blockchain trilemma"*, illustrated in Fig. 3, which is essentially the reformulation of the PACELC theorem for the blockchain domain [20]. 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

[2]The CAP acronym stands for *Consistency, Availability* and *Partition-tolerance*

[3]More on Bitcoin Confirmations at https://en.bitcoin.it/wiki/Confirmation

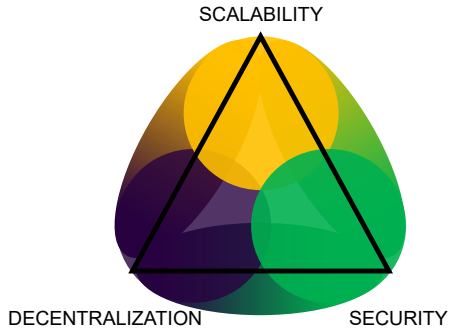

Figure 3. The blockchain trilemma is illustrated by a triangle: a feasible point cannot be close to all corners, meaning that a trade-off among the three properties must be chosen.

difficulty to speed up the block generation rate. However, this would compromise security, since less computing power becomes sufficient to perform a successful attack. Another strategy could be to change the trust model, for example, by 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 at the same time [20].

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.

IV. Towards a Blockchain Definition

We have discussed the structure of the blockchain (Sect. II) and distributed consensus protocols (Sect. III), stressing on the limits and trade-offs inherent to the blockchain technology. However, the literature of applications of the blockchain (mentioned in the introduction) make it seems like an almost universal, limitless technology. We argue that this apparent universality is rooted in the ambiguity of the *blockchain* term itself.

To clarify its possible usage we first need to formulate an unambiguous definition for the term "blockchain." To this end, thanks to Fig. 4 we compare the competing technologies for the implementation of a Shared Ledger with the most popular platforms commonly considered as blockchains, looking for the distinguishing features that will constitute our blockchain definition. Fig. 4 compares therefore two traditional DB technologies with the yet undefined concept of "Classic Blockchains", which captures all of those platforms (such as Ethereum and Monero) that preserved the distinctive features first introduced by Bitcoin.

A. Blockchain vs. Traditional Technologies

1) Centrally Managed DBs: They are maintained by a central administrator 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.

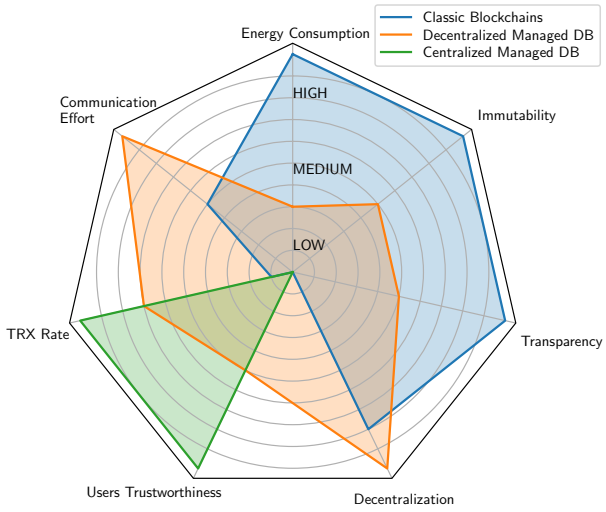

Figure 4. Multidimensional comparison of the blockchain with traditional Shared Ledger technologies: i.e., centralized and decentralized managed DBs.

The greatest advantage of one such implementation is the high level of efficiency in terms of transaction rate, communication effort and 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 as the 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.

2) Distributedly Managed DBs: These are cooperatively maintained by a group of administrators, and 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. 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 to achieve 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).

3) Classic Blockchains: Represented by Bitcoin, Ethereum and by all the other PoW-based blockchains[4] that, together, account for more than 90% of the total market capitalization of existing digital cryptocurrencies [21].

[4]Examples of other famous PoW-based cryptocurrencies are Monero, Bitcoin-Cash, Litecoin, Namecoin, Dogecoin, Primecoin, Auroracoin, Etherum-Classic and Zcash.

Classic blockchains turn 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.

B. Connotative Definition of Blockchain

The analysis of the competing technologies for the implementation of a Shared Ledger suggests what are the distinctive characteristics of a blockchain that distinguish it from all other Shared Ledger technologies. We condense these characteristics in the following definition of the term "blockchain":

Def. IV.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 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, then a centralized trusted registrar—potentially discriminatory—would become necessary, compromising the ledger decentralization. The openness to anonymous users 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 otherwise the distributed validation of transactions becomes impossible. The solution offered by blockchains is to record the PUBLIC & FULL HISTORY OF TRANSACTIONS and to safeguard it with a STRONG DISTRIBUTED CONSENSUS PROTOCOL. This is why a mechanism like the PoW, that makes historical blocks immutable and is mathematically secure regardless of any trust assumption on users, is a distinctive element of blockchains. The arguments we used to justify our definition of blockchain are concisely summarized in Highlight IV.1.

Highlight IV.1: Arguments supporting Def. IV.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 protocol protects from falsification.

C. Comparison with other definitions

In computer science, a first definition of a blockchain can be restricted to the simple data structure made of blocks of information chained by hash pointers [22]. 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 this wider definition: *"[The] blockchain is an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way"* [23].

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, however, it does without being axiomatic. Rather, it acknowledges the blockchain as an open, verifiable and immutable technology only by derivation of these properties from the three inseparable elements that together constitute the essence of a blockchain. Definition IV.1 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. Definition IV.1 also serves our clarification purpose, being the base for claiming that *the blockchain is not a universal technology*. Instead, given its characteristics, the blockchain is advantageous for a limited number of applications only.

D. Permissioned Ledgers are Blockchains?

Definition IV.1 raises a question: since permissioned ledgers are not openly verifiable, nor safeguarded by a strong consensus protocol, shall we call them blockchains?

1) 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, enterprises need their business-critical transactions to be kept confidential: their ledgers are therefore opaque, not verifiable by any external agent. For these reasons permissioned ledgers are not a truly decentralized nor transparent technology.

2) Less Immutable means less Secure: A trust model with registered/permissioned users is certainly safer than a model where users are anonymous. Strengthened by stronger 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.

3) Permissioned Platforms are Traditional Ledgers: Permissioned platforms seem to be not much different from traditional ledgers that existed also before Bitcoin [24], as they are empowered by traditional consensus protocols and their trust model still depends on a central authority.

Fig. 5 depicts our vision of the landscape of Shared Ledger technologies, with the blockchain positioned according to Definition IV.1 as provided in Sect. IV-B.

E. Proof of Work or Proof of Stake?

Many popular blockchain systems (in primis Ethereum) are planning the transition from PoW to PoS to stop wasting

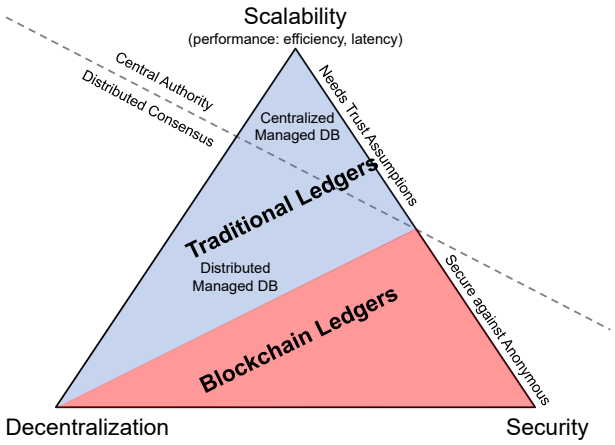

Figure 5. Position of the Blockchain in the landscape of the Shared Ledger technologies.

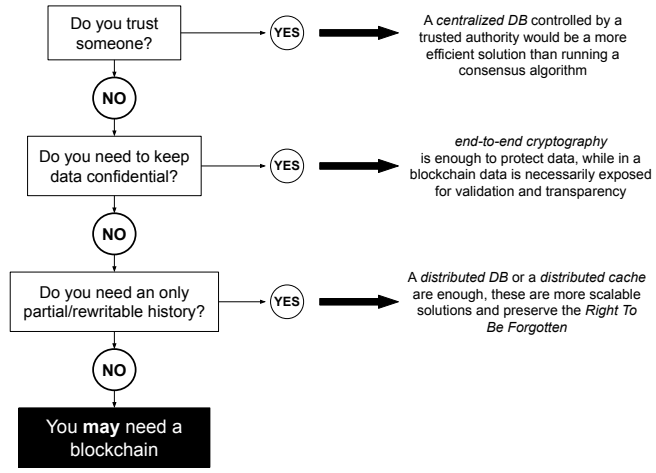

Figure 6. Application requirements and ledger technology: Aid to decision.

computing power, but will this transition compromise the blockchain characteristics? There is a hot debate in the literature on this topic, and we refer the interested user to Section 4 in [13], jumping here directly to the conclusions. We claim that both PoW and PoS 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 large amount of energy, but with the risk of long term instability.

However, there is 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 Proof of Stake (PoS)-system only depends on speculative mechanisms. Therefore, while with PoW the cost of an attack is predictable given the amount of the total computing power available, 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 and can change abruptly following the price of the stake.

V. Blockchain in Support of the IoT

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 acknowledges it as extremely inefficient [25], [12]. 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 Fig. 6, which extends the tradition started by Peck and Wüst [26], [27] to provide a chart guiding developers in the selection of the appropriate blockchain for their application. Our chart distinguishes itself from previous ones for its limited scope, i.e., it focuses on the cases when the blockchain is *not needed*. Fig. 6 also highlights, implicitly, those recurrent abuses of the blockchain in applications whose requirements conflict with the essential blockchain characteristics. In the extended version of this position paper [13] we dedicate indeed an

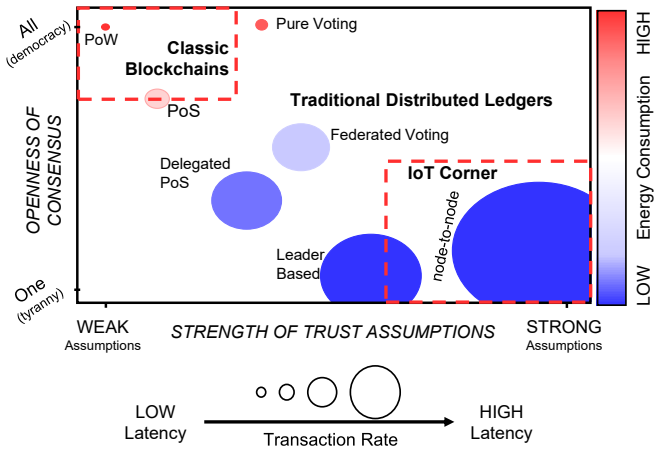

Figure 7. Bubblechart comparing the most popular consensus fabrics by trustworthiness of users (x-axis) and openness of the consensus protocol (y-axis). The color and the size of each bubble offer a quick indication of resource consumption and transaction rate.

entire section to the analysis of such abuses that we name *"The blockchain pitfalls."*

Although the analysis we did so far hints to a number of incompatibilities between the definition we constructed of blockchain and its application within the IoT, here we want to convey a constructive criticism to the blockchain, indicating how it can still be used as an *external service* in support of the decentralized validation of IoT transactions, offering a complementary or alternative paradigm to centralized cloud services.

To understand how the blockchain can successfully play as an external —not integrated— ledger for the IoT, consider the "bubblechart" of Fig. 7, which draws a multidimensional overview of the most popular consensus mechanisms according to the following four dimensions:

- The strength of the trust assumption, 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 (y-axis);
- Resource consumption (color of each bubble: Red when high or blue if low);
- The transaction rate (size of each bubble: The larger, the faster).

The figure enriches the trilemma by breaking down the

"scalability vertex" (Fig. 3) into two distinct dimensions, i.e., resource consumption and transactions rate. The ideal "blockchain-for-IoT" bubble would be a blue and large one in the top-left corner of the chart, thus being low-power, very performing, fully decentralized and extremely secure. Blockchain systems are naturally located in this top-left corner, characterized by being democratically open and secure despite weak trust assumptions, however they are slow and resource-hungry. The opposite corner is where IoT applications actually reside, with their scalability requirements, tight resource constraints, and high global transaction rates. This corner also highlights that the ultimate participants are "things" rather than humans.

Despite the lack of space for them in the bottom-right corner of our bubblechart, blockchains can still play an important role if we consider *node-to-node consensus* as a means to build trust. The key word here is *node-to-node*, which restricts 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 a 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" both from the world of cryptocurrencies (*Networks of Payment Channels* [28] and from that of communications, where a network is a set of channels interconnecting its nodes.

A. 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. A Transaction Channel is therefore a node-to-node consensus protocol where the two transacting parties establish a fast settling 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, and what is stored in the blockchain is a summary 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 [29], which scales up the technique to a full network of such channels. 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 [30]. 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 [31], [32] and rebalancing techniques [33].

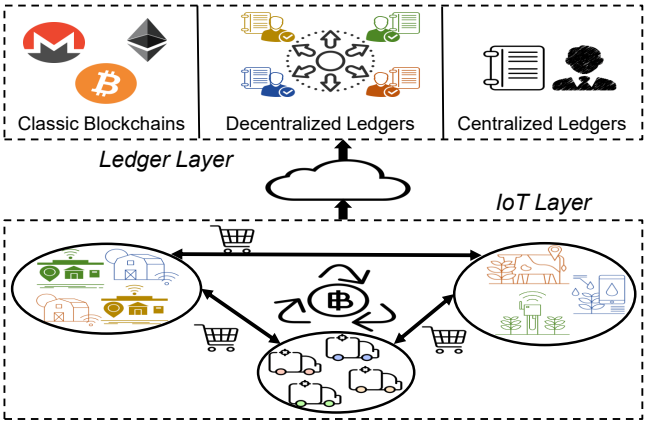

Figure 8. Different IoT clusters, made of devices managed either by private (home/enterprises) or public (institutional) entities perform most transactions locally, in the IoT layer. For interoperability purposes the different clusters access intermediary platforms. At this stage blockchain-based ledgers can provide a unique service independent of any trusted authority.

B. The Role of the Blockchain in the IoT

Fig. 8 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 have a *local* span first, from domotics to precision farming, to any industrial application. Industrial IoT, furthermore, 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 can 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.

VI. Conclusions

This position paper argues that the blockchain is not an appropriate technology for integration in the IoT, but 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.

Starting from the theoretical boundaries set by consensus protocols, we raise the concern that stake-based protocols fully rely on the rationality assumption of their users and 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.

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.

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 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] N. Kshetri and J. Voas, "Blockchain-Enabled E-Voting," *IEEE Software*, vol. 35, no. 4, pp. 95–99, jul 2018.

[3] 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 Jou.*, vol. 6, no. 5, pp. 7992–8004, oct 2019.

[4] 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 Jou.*, vol. 6, no. 5, pp. 8770–8781, oct 2019.

[5] Y. Guo and C. Liang, "Blockchain application and outlook in the banking industry," *Springer Financial Innovation*, vol. 2, no. 1, dec 2016.

[6] L. Ghiro, 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.

[7] 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 Jou.*, vol. 6, no. 5, pp. 7702–7712, oct 2019.

[8] 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.

[9] 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.

[10] 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 Jou.*, vol. 6, no. 5, pp. 8114–8154, oct 2019.

[11] 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 Jou.*, vol. 6, no. 5, pp. 8155–8168, oct 2019.

[12] A. de Vries, "Bitcoin's Growing Energy Problem," *Elsevier Joule*, vol. 2, no. 5, pp. 801–805, may 2018.

[13] L. Ghiro, F. Restuccia, S. D'Oro, S. Basagni, T. Melodia, L. Maccari, and R. Lo Cigno, "What is a Blockchain? A Definition to Clarify the Role of the Blockchain in the Internet of Things," feb 2021. [Online]. Available: https://arxiv.org/abs/2102.03750

[14] 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.

[15] N. T. Courtois, "On The Longest Chain Rule and Programmed Self-Destruction of Crypto Currencies," 2014. [Online]. Available: https://arxiv.org/abs/1405.0534v11

[16] 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.

[17] S. M., J. Spaulding, L. Njilla, C. Kamhoua, S. Shetty, D. Nyang, and D. Mohaisen, "Exploring the Attack Surface of Blockchain: A Comprehensive Survey," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 3, pp. 1977–2008, 2020.

[18] 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.

[19] 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.

[20] 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.

[21] F. Armknecht, J.-M. Bohli, G. O. Karame, and W. Li, "Sharding PoW-based Blockchains via Proofs ofKnowledge," *IACR Cryptol. ePrint Arch.*, vol. 2017, p. 1067, 2017.

[22] C. Halatsis and G. Philokyprou, "Pseudochaining in Hash Tables," *Communications of the ACM*, vol. 21, no. 7, pp. 554–557, jul 1978.

[23] M. Iansiti and K. R. Lakhani, "The Truth About Blockchain," *Harvard Business Review*, pp. 118–127, jan 2017.

[24] 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: April 2021].

[25] P. Fairley, "Feeding the Blockchain Beast," *IEEE Spectrum*, vol. 54, pp. 36–59, oct 2017.

[26] 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.

[27] 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.

[28] 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.

[29] 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: April 2021].

[30] V. Sivaraman, S. B. Venkatakrishnan, K. Ruan, P. Negi, L. Yang, R. Mittal, G. Fanti, and M. Alizadeh, "High Throughput Cryptocurrency Routing in Payment Channel Networks," in *Proc. of the 17th USENIX Symp. on Networked Syst. Design & Implementation (NSDI'20)*, Santa Clara, CA, feb 2020, pp. 777–796.

[31] 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.

[32] L. Maccari, L. Ghiro, 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.

[33] R. Pickhardt and M. Nowostawski, "Imbalance measure and proactive channel rebalancing algorithm for the Lightning Network," in *Proc. of the IEEE Int. Conf. on Blockchain & Cryptocurrency (ICBC'20)*, may 2020.