LEDGER
ledgerjournal.org

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

REVIEW ARTICLE

A Taxonomy of Blockchain Technologies: Principles of Identification and Classification

Paolo Tasca,* † Claudio J. Tessone‡

Abstract. A comparative study across the most widely known blockchain technologies is conducted with a bottom-up approach. Blockchains are deconstructed into their building blocks. Each building block is then hierarchically classified into main and subcomponents. Then, varieties of the subcomponents are identified and compared. A taxonomy tree is used to summarise the study and provide a navigation tool across different blockchain architectural configurations.

1. Introduction

1.1. Background—The original combination of a set of existing technologies (distributed ledgers, public-key encryption, merkle tree hashing, consensus protocols) gave origin to the peer-validated decentralised cryptocurrency called Bitcoin, originally introduced by Satoshi Nakamoto, in 2008.[1] That year was the advent of a new technological milestone: the *blockchain*. Indeed, blockchains have an impact well beyond the specific case of Bitcoin. Blockchains allow new forms of distributed software architecture to be developed where networks of untrusted (and sometimes even "corrupted") participants can establish agreements on shared states for decentralised and transactional data in a secure way and without the need for a central point of control or regulatory supervision. Blockchains ensure trust among anonymous counterparts in decentralised systems without the need of central supervisory authorities in charge of verifying the correctness of the records in the ledger. Blockchains were announced as a disruptive technological innovation, but in fact there is no true technical innovation in Bitcoin and blockchain. All the components had already been developed before the publication of the Bitcoin whitepaper in 2008. From a historical perspective, the technology has its roots in Ralph C. Merkle's elaborations, who proposed the Merkle Tree—the use of concatenated hashes in a tree for digital signatures—in the 1970s. Hashing has been used since the 1950s for cryptography for information security, digital signatures, and message-integrity verification. A decade later, Leslie Lamport proposed using the hash chain for a secure login. The first cryptocurrency for electronic cash was described at the dawn of the web in 1990. Further evolutions and refinements of the hash chain concept were introduced in a paper by Neil Haller on the S/KEY application of a hash chain for Unix login,

* 3ELK15Knpz3bAB4t4VquvrW8gzi1oHMwXv

† P. Tasca (p.tasca@ucl.ac.uk) is founder and Executive Director of the Centre for Blockchain Technologies at the University College London, UK.

‡ C. J. Tessone (claudio.tessone@business.uzh.ch) is Assistant Professor of Network Science at the University of Zurich and co-founder of the UZH Blockchain Centre.

in 1994. While Adam Back proposed hashcash in 2002, the first eletronic currency based on a blockchain with the Proof-of-Work (PoW) concept was proposed in Nakamoto's disruptive paper.[1] While blockchains are still in their emergent technological phase, they are fast evolving with the potential to see applications in many sectors of our socio-economic systems. According to some statistics summarised by the World Economic Forum, the interest in blockchains has expanded globally.[2] Almost thirty countries are currently investing in blockchain projects. In the finance sector, 80% of the banks predict to initiate blockchain-related projects by 2017. Additionally, venture capital investments with a focus on blockchain activities grew to over 1.4 billion USD from 2014 to 2016. Since blockchain-based digital currencies combine together features of money with those of a payment system,[3] central banks have also started to look into the technology. Currently, over nineteen central banks worldwide do blockchain research and development. Some of them—*e.g.*, the Bank of England—have already commissioned studies on CBDCs (Central Bank Digital Currencies). From the industry side, over one hundred corporations have joined blockchain working groups or consortia and the number of patents filled has risen to more than three thousand at the time of writing. These figures show the importance and awareness of blockchain as one of the most promising emerging technologies that—together with Artificial Intelligence, Internet of Things, and nanotechnology—will have a pervasive impact on the future of our society.

1.2. Problem Statement and Research Method—At the time of writing, we may argue that—according to the Technology Life Cycle theory—we are at the beginning of the so called phase of "fermentation" which is characterised by technological uncertainty due to the evolution of blockchains into alternative technological paths. The industry promotes different model designs favouring functional and performance aspects over others in order to meet specific business goals. At present, there are thousands of blockchain-based projects worldwide under development. Some of them are simple forks of successful technologies, such as Bitcoin or Ethereum; others propose completely new functionalities and architectures. It is for this reason that, instead of "the Blockchain," we refer to *blockchains* or *blockchain technologies*, that is, in order to encompass all the possible architectural configurations and, for the sake of simplicity, also the larger family of distributed ledger technologies (DLT) (*i.e.*, community consensus-based distributed ledgers where the storage of data is not based on chains of blocks). A heterogeneous development combined with a lack of interoperability may endanger a wide and uniform adoption of blockchains in our techno- and socio-economic systems. Moreover, the variations in blockchain designs and their possible configurations represent a hindrance for software architectures and developers. In fact, without the possibility to resort to a technical reference model, it is difficult to measure and compare the quality and performance of different blockchains and applications that sit atop them. To summarise, current variations in blockchain software architectures pose a number of concerns from different perspectives, specifically when it comes to heterogeneity. Heterogeneity is a problem for the future development of blockchain technologies, because it will prevent their development, adoption, and stimulation of innovation. Just a few of the problems it may create include:

(1) Preventing consistency in the drafting of laws and policies related to the regulation of blockchains/DLT.
(2) Increasing ambiguity in the application of consumer protection laws and regulations.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

(3) Decreasing the predictability of how the workforce may be affected by blockchains/DLT.
(4) Decreasing the accuracy of academic research into the concepts that underpin the development of new applications and solutions.
(5) Hindering the development and use of solutions using blockchains and DLT for ISO, IEC, and other SDOs.
(6) Increasing the complexity of understanding the uses of blockchains/DLT for NGOs and how these technologies may be applied in the relevant sectors to achieve social and economic goals

The solution to these problems is the setting up of software reference architectures where standardised structures and respective elements and relations can provide templates for concrete blockchain architectures. Standards can emerge naturally because of market adoption (industry-driven) or can be imposed by institutes and organisations. In the first group we may include initiatives like the Accord Project (https://www.accordproject.org/), the ChinaLedger (http://www.chinaledger.com/) or R3 (https://www.r3.com/). In the second group we may refer to the initiative conducted by the International Organization for Standardization (ISO) with the establishment of the technical committee ISO/TC 307 on Blockchain and Distributed Ledger Technologies. Several working groups have been set up with different topics to discuss. For just one example, the ISO/TC 307/WG1 working group is engaged with reference architecture, taxonomy, and ontology. Overall, a long-term standardisation of blockchain reference architecture will benefit every industry. A standard for software reference architecture is necessary in order to enable a level playing field where every industry player and community member can design and adopt blockchain-enabled products or services under the same conditions with possibility of data exchange. As it is for the Internet, several institutes of standardisations (*e.g.*, ETF in cooperation with the W3C, ISO/IEC, ITU) set a body of standards. Internet standards promote interoperability of systems on the Internet by defining precise protocols, message formats, schemas, and languages. As a result, different hardware and software can seamlessly interact and work together. Applied to the World Wide Web (as a layer on the top of the Internet), standards bring interoperability, accessibility and usability of web pages. Similarly, the adoption of blockchain standards will promote the blossoming and proliferation of interoperable blockchain-enabled applications. Thus, if we envisage a future in which blockchains will be one of the pillars of our societies' development, it is necessary to begin discussing and identifying standards for blockchain reference architectures. The aim of this study is to highlight the need for standard technical reference models of blockchain architectures. This is conveniently aligned with the industry sentiment which currently pushes organisations for standardisation to set industry standards. In order to support an appropriate co-regulatory framework for blockchain-related industries, a multi-party approach is necessary as it is for the Internet where both national standards, international standards, and a mixture of standards and regulation are in place. In the mid-to-long term, the lack of standards could bring risks related to privacy, security, governance, interoperability, and risk to users and market participants, which could appear in the form of blockchain-related cyber crimes. From a preliminary survey conducted in 2016 by Standards Australia, more than 88% of respondents supported a role for standards in supporting the roll out of blockchain technologies.[4] Given the above problems, the goal of this research is to conduct a review of the blockchain literature. This will be a preparatory work in order to identify and logically group different blockchain (main and sub) components and their layouts. In order to achieve our goal we propose a blockchain

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

taxonomy. Taxonomy comes from the term "taxon" which means a group of organisms. In our case, taxonomy encompasses the identification, description, nomenclature, and hierarchical classification of blockchain components. This is different from an ontology which would be more focused on the study of the types, properties, and interrelationships of the components and events that characterize a blockchain system. The methodological approach is composed of the following steps:

(1) Analysis across blockchains. A precondition is the analysis of vocabulary and terms to sort out ambiguities and disagreements. A literature review of the existing technologies is the starting point to limit complexity and organise information in schematic order. To avoid ambiguities the analysis is supported by a merge of common blockchain terminologies developed so far in the literature and grouped in an online database.[5] This brings together a vocabulary of key blockchain terms to provide readers with a foundation upon which to understand the classification and taxonomy developed in the rest of the analysis. The identification of the blockchain components is a crucial part of this analysis. In order to explore all the possible domains of blockchain components and their topological layout indicating their runtime interrelationships, we conduct a comparative study across different families of blockchain applications: digital currencies, application stacks, asset registry technologies, and asset-centric technologies. See Table 1 in Appendix and Tasca, "Digital Currencies" (2015) for more information.[6]

(2) Framework setting. After the comparative study, a hierarchical taxonomy (a tree structure of classifications for a given set of components) has been defined and populated with *main*, *sub* and (when necessary) *sub-sub* components.

(3) Layout categorisation. Finally, for the components in the lowest level of the hierarchical structure, different layouts are introduced and compared. However, as the technology keeps evolving, the layouts are increasing over time. Thus, for the sake of simplicity, we limit our study to two or three main layouts per each *sub* or *sub-sub* component.

1.3. Results—The result of the component-level analysis is a universal blockchain taxonomy tree that groups the major components in a hierarchical structure, and identifies their functional relation and possible design patterns.

In general, it is difficult to evaluate whether a taxonomy or an ontology is good or bad, especially if the domain is a moving target like blockchains. Taxonomies and ontologies are generally developed to limit complexity and organise information, but all serve different purposes and generally evolve over time (see *e.g.*, the evolution of the famous Linnaean taxonomy in biology). Thus, our taxonomy simply aims to contribute to set the foundations for classifying different kinds of blockchain components. Without claiming to represent the ultimate structure, the proposed taxonomy could be of practical importance in many cases. For example, it can:

(1) support software architectures to explore different system designs and to evaluate and compare different design options;

(2) be propaedeutic to the development of blockchain standards with the aim to increase the adoption at a large scale of blockchain-enabled solutions and services;

(3) enable research into architectural frameworks for blockchain-based systems in order to boost the adoption of blockhain-enabled systems, through their interoperability and compatibility;

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

(4) create gateway models to multiple blockchains and design governance frameworks;
(5) promote blockchain predictability;
(6) aid the development of regulatory frameworks that could provide a mix of both legal and technical rules (*i.e.*, regetech for blockchain-based systems).[7]

2. Background on Blockchain Technologies

Since Bitcoin's inception in 2009, many blockchain software architectures have been deployed to meet different technical, business, and legal design options. Given the current complex dynamic of blockchain architectural development, it would be neither exhaustive nor comprehensive to provide a picture of the existing blockchain technologies developed so far. Therefore, we take a bird's-eye view and describe blockchains by looking at their key driving principles, such as data decentralisation, transparency, security, immutability, and privacy.[8]

Decentralisation of consensus. The distributed nature of the network requires untrusted participants to reach a consensus. In a blockchain, consensus can be based on "rules" (that determine, *e.g.*, which transactions are allowed and which are not, the amount of bitcoins included in the block reward, the mining difficulty, etc.) or on the history of "transactions" (which allows users to determine who owns what). The decentralised consensus on transactions governs the update of the ledger by transferring the responsibilities to local nodes which independently verify the transactions and add them to the most cumulative computation throughput (*i.e.*, the "longest chain" rule). There is no integration point or central authority required to approve transactions and set rules. No single point of trust and no single point of failure.

Transparency. Records are auditable by a predefined set of participants, although the set can be more or less open. For example, in public blockchains everyone with an Internet connection to the network holds equal rights and ability to access the ledger. The records are thus transparent and traceable. Moreover, network participants can exercise their individual (weighted) rights (*e.g.*, measured in CPU computing power) to update the ledger. Participants have also the option to pool together their individual weighted rights.

Security. Blockchains are shared, tamper-proof, replicated ledgers where records are irreversible and cannot be forged thanks to one-way cryptographic hash functions. Although security is a relative concept, we can say that blockchains are relatively secure because users can transfer data only if they posses a private key. Private keys are used to generate a signature for each blockchain transaction a user sends out. This signature is used to confirm that the transaction has come from the user, and also to prevent the transaction from being altered by anyone once it has been issued.

Immutability. Blockchains function under the principle of the non-repudiation and irreversibility of records. Blockchains are immutable because once data has been recorded in the ledger, it cannot be secretly altered ex-post without letting the network know it (*i.e.*, their data is tamper-resistant). In the blockchain context immutability is preserved thanks to the use of hashes (a type of a mathematical function which turns any type of input data into a fingerprint of

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

fixed size, that data called a hash. If the input data changes even slightly, the hash changes in an unpredictable way) and often of blocks. Each block includes the previous block's hash as part of its data, creating a chain of blocks. The strength of a given blockchain's immutability is relative, and relates to how hard the history of transactions is to change. Indeed, it becomes very difficult for an individual or any group of individuals to tamper with the ledger, unless these individuals control the majority of "voters." For public proof-of-work blockchains such as Bitcoin, the immutability is related to the cost of implementing a so-called "51% attack." For private blockchains, the block-adding mechanism tends to be a little different, and instead of relying on expensive Proof-of-Work, the blockchain is only valid and accepted if the blocks are signed by a defined set of participants. This means that, in order to recreate the chain, one would need to know private keys from the other block-adders. A complete discussion of threats to the immutability of a given transaction history can be found in Barber (*et al.*) "Bitter to Better" (2012).[9] While blockchains are technically immutable, from a governance perspective, this immutability is never fully realised. There have been several examples where the Bitcoin community has reverted Bitcoin blocks based on community decisions. The division between Ethereum and Ethereum classic, and later between Bitcoin and Bitcoin Cash and Bitcoin Gold are not purely anecdotal evidence: they are strong indicators of the importance that governing bodies—even if informal—end up having on the information eventually stored in a blockchain.[10]

Other non-fundamental properties of blockchains include data automation and data storage capacity.

Automation and smart contracts. Without the need for human interaction, verification, or arbitration, the software is written so that conflicting or double transactions are not permanently written in the blockchain. Any conflict is automatically reconciled and each valid transaction is added only once (no double entries). Moreover, automation regards also the development and deployment of smart legal contracts (or smart contract codes, see Clack, Bakshi, and Braine, "Smart Contract Templates" (2016)[11]) with payoff depending on algorithms which are self-executable, self-enforceable, self-verifiable, and self-constrained.

Storage. The storage space available on blockchain networks can be used for the storage and exchange of arbitrary data structures. The storage of the data can have some size limitations placed to avoid the "blockchain bloat" problem.[12] For example, metadata can be used to issue meta-coins: second-layer systems that exploit the portability of the underlying coin used only as "fuel." Any transaction in the second layer represents a transaction in the underlying network. Alternatively, the storage of additional data can occur "off-chain" via a private cloud on the client's infrastructure or on public (P2P or third-party) storage. Some blockchains, like Ethereum, also allow the storage of data as a variable of smart contracts or as a smart contract log event.

3. Taxonomy of Blockchains

The diversity of blockchain research and development provides an opportunity for cross-fertilisation of ideas and creativity, but it can also result in fragmentation of the field and duplication of efforts. One solution is to establish standardised architectures to map the field

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

and promote coordinated research and development initiatives. However, in terms of blockchain software architecture design, little has been proposed so far,[13] and the problem of consistently engineering large, complex blockchain systems remains largely unsolved. We approach this problem by proposing a component-based blockchain taxonomy starting from a coarse-grained connector–component analysis. The taxonomy compartmentalises the blockchain connectors/-components and establishes the relationships between them in a hierarchical manner. We adopt a reverse-engineering approach to unbundle the blockchains and divide them into *main* (coarse-grained) components. Each main component is then split into more (fine-grained) *sub* and *sub-sub* components (where necessary). For each of these sub (and/or sub-sub) components, different *layouts* (models) are identified and compared. By deriving the logical relation between (main, sub or sub-sub) components, the study helps to clarify the alternative *modus operandi* of the blockchains, and helps to develop the conceptual blockchain design and modelling.

Similarly to other fields, like electronics or mechanics,[14] the software engineering approach used to derive the taxonomy treats blockchains as the result of gluing together prefabricated, well-defined, yet interdependent components. Although equivalent components provide similar services and functions, they can be of different importance and type, and the interconnection may work in different ways. Following this logic, each of the next seven sections will introduce a new blockchain main component and its sub (and eventually sub-sub) components by describing and comparing their layouts.

4. Consensus

The first identified main component is *Consensus*. It relates to the set of rules and mechanics that allows for the maintainance and updating of the ledger and guarantees the trustworthiness of the records in it, *i.e.*, their reliability, authenticity, and accuracy.[15] Consensus varies across different blockchain technologies, with every consensus mechanism bringing advantages and disadvantages based on different characteristics *e.g.*, transaction speed, energy efficiency, scalability, censorship-resistence, and tamper resistance.[16] The set of rules and mechanics compose the framework of the validation process that is necessary to overcome security issues during the validation. Figure 1 illustrates the subcomponents forming the component Consensus:

- [1] Consensus Network Topology
- [2] Consensus Immutability and Failure Tolerance
- [3] Gossiping
- [4] Consensus Agreement
  - [4.1] Latency
  - [4.2] Finality

Those subcomponents and sub-subcomponents must be jointly considered when designing an active network consensus validation process, because not only their individual configuration but also their combination determine when and how the overall blockchain agreement is achieved and the ledger updated.

4.1. Consensus Network Topology—Consensus Network Topology describes the type of interconnection between the nodes and the type of information flow between them for transaction and/or for the purpose of validation. For efficiency reasons, systems have historically been

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

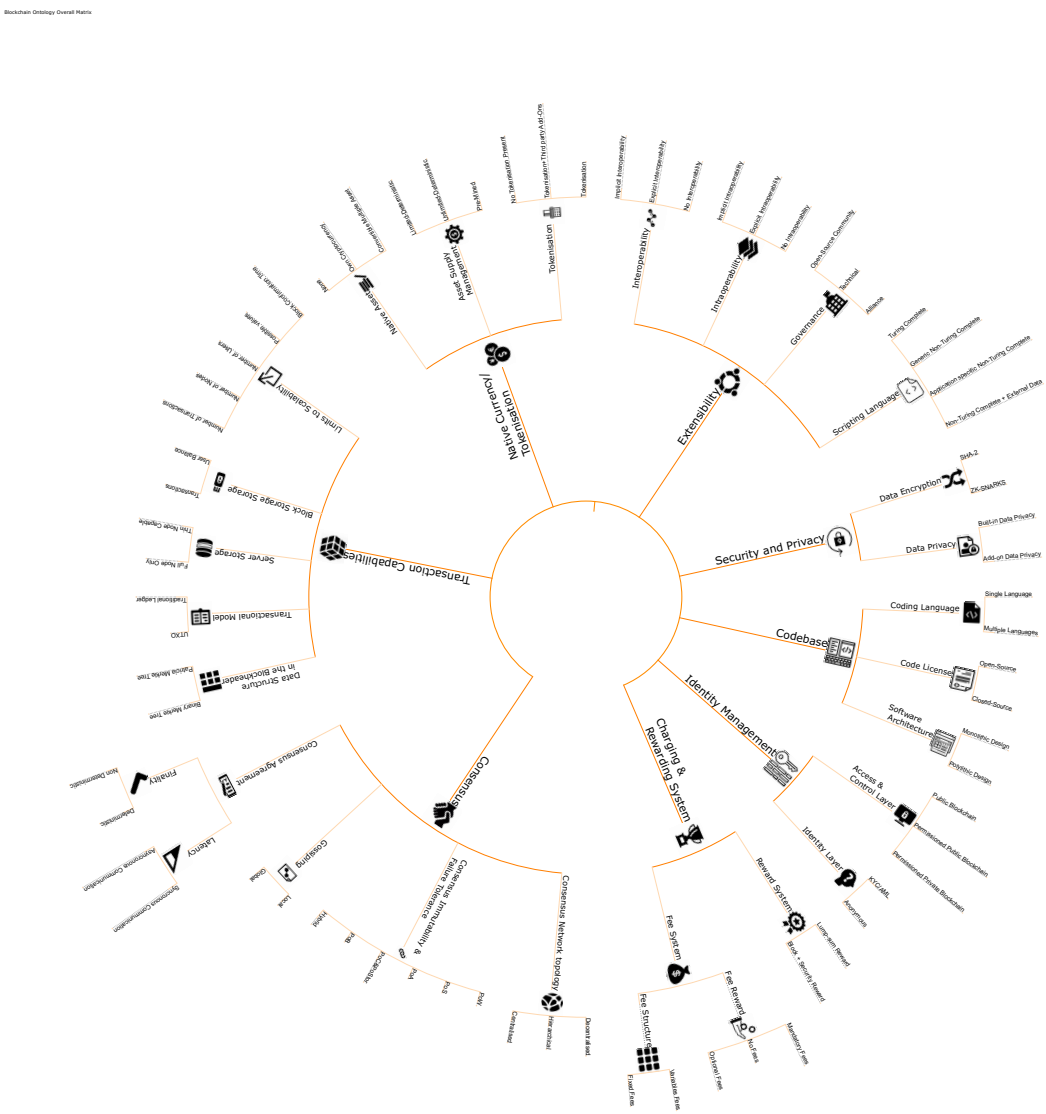

Fig. 1. Blockchain Taxonomy Tree: A representation of the taxonomic deconstruction of blockchain-based technologies.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

designed in a centralised manner. This centralisation lowers dramatically the costs for system configuration, maintenance, and adjustment (and the costs of arbitration in case of conflict) as this work has to be performed only once in a central place. While highly efficient in many situations, this kind of system creates a single (or very limited set of) point(s) of failure and suffers from scalability issues. With respect to the network topology, this hierarchical arrangement is still present in most of our techno- and socio-economic systems (one example is the modern electronic payment system). To avoid the single point of failure, these centralised systems can be extended into hierarchical constructs which exhibit larger scalability and more redundancy, while keeping the communication efficient. Decentralised solutions have been proposed as a alternatives to those centralised topologies. Since the dawn of the Internet, technical systems have evinced a transition towards decentralised arrangements where all the nodes are equivalent to any other.[17] For most applications, blockchain-based systems—with their federated sets of participants—are a clear example of this kind. Blockchain-based systems resort to specific topologies to create the peer network that ultimately determines how the validation process will evolve.

It is important to mention that Consensus Network Topology is linked to the level of (de)centralisation in the validation process, but this is not the only determinant. Also, other factors like the reward mechanism (see Section 11) heavily influence the validation.[15] As a matter of an example, Bitcoin has a decentralised validation process, which is still accompanied by an ever-increasing concentration of computational power devoted to the Proof-of-Work by network participants. Indeed, during the period 2013-2015, the cumulative market share of the largest ten pools relative to the total market hovered in the 70% to 80% range.[6]

We identify three possible layouts for Consensus Network Topology:

(1) **Decentralised.** There exist implementations that are decentralised. Bitcoin as the pioneer in digital currencies established a distributed P2P network, which enables direct transactions between every node within the network. The validation process within the Bitcoin network is decentralised through miners and full nodes who validate the transactions within the network connected in a random way, as provided by super-nodes.[1] This network illustrates a decentralised Consensus Network Topology. Obviously, this layout is independent from the *Consensus Immutability* layout (Peercoin and NXT also show decentralised network topologies).

(2) **Hierarchical.** There are other implementations that are not decentralised, and there exists differencces in the roles nodes have. For example, in Ripple the network topology is divided into tracking (or stock) and validating nodes.[18] The tracking nodes are the gateway for submitting a transaction or executing queries for the ledger. They make also available the transactions to the broader network via history sharding.[19] In addition to that the validating nodes have the same functions as tracking nodes, but they can also contribute additional sequences to the ledger by validation, vote on fees and amendments.[20] This Kind of solution yields (or can be extrapolated to) a hierarchical network topology. In the community of developers these hierarchical topologies are also referred to as "Consortium blockchains."

(3) **Centralised.** In some specific implementations, a central authority may need (or wish) to control what is added to the ledger. An example for this are digital versions of fiat currencies: the so-called Central Bank Digital Currencies (SBDCs). This kind of solution yields a third layer, "Centralised topology," which is intimately related to private

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

blockchains. It is important to mention that a centralised solution would normally be considered a non-properly working design (or a non-solution) if implemented in terms of a blockchain, as it would have been implemented in a more transparent manner otherwise. Normally, some level of federation and redundancy are key to blockchain systems.

4.2. Consensus Immutability and Failure Tolerance—In general, the failure tolerance of a distributed system should be defined with respect to three interrelated issues: faults (*e.g.*, Byzantine faults), errors, and failures. See *e.g.*, Driscoll (*et al.*) "Byzantine Fault Tolerance" (2003) and Castro (*et al.*) "Practical Byzantine Fault Tolerance" (1999) for Byzantine fault tolerance in distributed systems.[21,22] There are different types of failures, and generally it is costly to implement a fault-tolerant system. Practically, it is not possible to devise an infallible, reliable system. For a literature review and deeper analysis of fault-tolerant distributed systems, we refer the reader to Cristian, "Understanding Fault-Tolerant Distributed Systems" (1991) and Fischer, "The Consensus Problem in Unreliable Distributed Systems" (1983).[23,24] A blockchain, as a special kind of distributed system, is fault-tolerant when it shows the ability to continue functioning: *i.e.*, it must grant reliability, validity, and security of the information stored in the ledger. Indeed, blockchains represent a decentralised solution to the problem of information storage which requires no central database but rather many duplicates, such that each server holds a *replica* of the ledger. Any new record is costly (often measured in terms of computational power) to add to the ledger, but cheap to have verified by peers. Therefore, a blockchain system is in need of an efficient consensus mechanism to ensure that every node has its original version of the full transaction history which is kept consistent with the other peers over time. In this regard, the immutability of achieved consensus differs with respect to the resources required to maintain large network security. In the past years, the evolution of blockchain technologies has been accompanied by the development of different mechanisms that help to keep the information contained in the ledger reliable, valid, and secure. The mechanisms for *Consensus Immutability*, together with the subcomponents of *Consensus Agreement*, determine the failure tolerance of a blockchain. As of the time of writing, we have identified six main layouts for Consensus Immutability and Failure Tolerance:

(1) **Proof-of-Work**. The most widely used cryptocurrency, Bitcoin, uses Proof-of-Work (PoW) to ensure the immutability of transaction records. In this setup, computing devices, usually called *miners*, connected to a peer-to-peer network, perform the task of validating the transactions proposed for addition to the complete record of existing—valid—transactions. The generation of a block that can be appended to the blockchain—rendering in this way valid all transactions included therein—requires solving the inversion of a cryptographic function, which can only be done by brute force. In PoW, the probability that a miner mines a new block depends on the ratio between the computational power he devotes to this task and the total instantaneous computational power by all miners connected to the network. Specifically, miners must find a solution to a one-way hash function by computing new hash values based on the combination of the previous hash values contained in the message, the new transactions in the block they create, and a nonce. The solution is such that the new hash value will start with a given number of zeros $\leq$ target. As of the time of writing, the mining process needs several requirements to be successful.[25] These include specialised hardware, which is needed to perform the computational tasks, and ever-increasing amounts of electricity to power the hardware.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

These computations are run by dedicated machines (ASICs) which are very expensive as well as resource-intensive, contributing to a large electricity footprint for cryptocurrency miners.[26] Due to this scheme, in the last years miners have agglomerated around mining pools.[27] A clear drawback of the PoW mechanism, then, is that the inherent inefficiency in resource requirements, and the large-scale investments needed, have led to long-term centralisation of the mining power. In late 2017, almost five quadrillion SHA256 computations were performed each second in the Bitcoin mining process. Regretfully, these computations do not have any practical or scientific relevance apart from ensuring that the process of block creation is costly, while maintaining that the process of block verification remains simple for peers.

Interestingly, when two adversaries coordinate, it is sufficient that each hold only around 25% of the total computing power to mount an attack.[28] In this layout, there exists the risk of monopoly mining, induced by large coordination of miners in a single mining pool, which continuously increases the expected payoff of others if they join said mining pool. In this hypothetical situation, said mining pool can censor specific transactions and dictate which transactions are accepted and which not.

In contrast, BFT consensus mechanisms tolerate at most $n/3$ corrupted nodes in the asynchronous communication protocol and even higher levels in the synchronicity case. Electricity consumption can be estimated around 0.1 to 1 W/GH corresponding to around 1GW of electricity consumed every second. For this reason, other developers within the area of blockchain technologies are continuously attempting to develop novel mechanisms to achieve an equivalent result. It is worth mentioning that some cryptocurrencies (*e.g.*, Primecoin) have tried to make the PoW a task that serves a useful aim (in said case, searching for long chains of prime numbers, or Cunningham chains).

(2) **Proof-of-Stake**. PoS links block generation to the proof of ownership of a certain amount of digital assets (*e.g.*, digital currencies) linked to the blockchain. The probability that a given *prover* is selected to verify the next block grows in relation to the share of assets that prover has within the system. The underlying assumption is that users with a large share of the system wealth are more likely to provide trustworthy information with respect to the verification process, and are therefore to be considered a more trustworthy validator.[16] Two alternative PoS methods have been devised. The first one is based on randomised block selection (used in *e.g.*, NXT and BlackCoin); it uses a calculation searching for the lowest hash together with the stake size; it is therefore somewhat deterministic and each node can independently determine the likelihood of being selected in a future round. An alternative scheme is the *coin-age*-based selection (used by *e.g.*, Peercoin, being actually the first one to be implemented in real world conditions) which combines randomisation with coin-age (a number derived from multiplying the amount of assets held by the prover and the length of time those assets have been held). Although PoS has the chance to solve two issues with PoW (risk of monopoly mining and resources wasted in the mining process), it is affected by the "nothing at stake" issue. Because there is little cost in working on several chains (unlike in PoW), one could abuse the system by voting for multiple blockchain-histories which would prevent the consensus from ever resolving (double spending). This problem can be addressed by Delegated Proof-of-Stake (DPoS), a generic term describing an evolution of the basic PoS consensus (utilised

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

in, *e.g.*, BitShares, Casper by Ethereum, and Tendermint) where blocks are forged by predetermined users delegated by the user who has the actual stake. These forgers are rewarded for their duty and are punished for malicious behaviour (such as participation in double-spending attacks). This principle of pre-authorised forgers is generalised by the Proof-of-Authority mechanism.

(3) **Proof-of-Authority**. In this case, participants are not asked to solve arbitrarily difficult mathematical problems like in PoW, but instead are asked to use a hard-configured set of "authorities" empowered to collaborate "trustlessly." Namely, some nodes are exclusively allowed to create new blocks and secure the blockchain. Typically, Proof-of-Authority (PoA) mechanisms fit well for consortium private networks where some preselected real entities (*i.e.*, the *authorities*) are allowed to control the content that is added to the public registry. Those nodes will receive a set of private keys that will be used to "sign" the new blocks, acting as *trusted signers*. Thus, every block (or header) that a client sees can be matched against the list of trusted signers. The challenges brought by PoA are related to: control of mining frequency, distribution of mining load (and opportunity) between the various signers, and maintenance of the the list of signers so as to be robust from malicious attacks even in presence of dynamic mutation of the trusted signers.

(4) **Proof-of-Capacity/Proof-of-Space and Proof-of-Storage**. PoC or PoSpace and PoStorage are implementations of the popular idea of "space as resource." Here the focus is not on the CPU cycles but on the amount of actual memory (non-volatile) space the prover must employ to compute the proof. Nodes are asked to allocate a significant volume of their hard drive space to mining instead of using CPU-bound space as in PoW. Miners are incentivised to devote hard-drive capacity as those who dedicate more disk space have a proportionally higher expectation of successfully mining a block and reaping the reward. The PoC makes use of hash trees to efficiently allow verification of a challenge without storing the tree. These schemes are more fair and green than PoW. The reason mainly comes from the lower variance of memory access times between machines and the lower energy cost achieved through the reduced number of computations required. Several practical implementations adopt the PoC consensus algorithm like Permacoin, SpaceMint, and Burstcoin, to cite just a few. PoC consists of an initialisation and subsequent execution between a prover P and a verifier V.[29] Rather than P proving to V that some amount of work has been completed, P proves to V that she has allocated some number of bytes of storage. After the initialisation phase, P is supposed to store some data F of size N. Instead, V only holds some small piece of information. At any later point in time V can initialise a proof execution phase, and at the end V outputs reject or accept. The PoC is in general defined by three quantities: N_0, N_1, and T; then, the miner shows that she either: 1) had access to at least N_0 storage between the initialisation and execution phases and at least N_1 space during the execution phase; or 2) used more than T time during the execution phase. Solutions to the "mining multiple chains" and "grinding blocks" problems of PoC algorithms have been proposed by Park (*et al.*) "Spacecoin" (2015), among others.[30] The Proof-of-Storage (PoS) mechanism is similar to PoC but the designated space in it is used by all participants as common cloud storage.[31]

(5) **Proof-of-Burn**. In Proof-of-Burn (PoB), miners must prove that they burned some digital assets. They do so by sending them (*e.g.*, digital currencies) to a verifiable unspendable

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

address belonging to them. Similarly to the PoS, PoB is intended to minimise the waste of resources generated by PoW. However, at the current stage, all PoB mechanisms function by burning PoW-mined digital currencies. This is therefore an expensive activity as the digital currencies once required to work as "fuel" in a PoB system cannot be recovered.[15] PoB can be used also to bootstrap a token off of another (see *e.g.*, Counterparty or Mastercoin).

(6) **Hybrid**. The more advanced hybrid consensus immutability and failure tolerance methods are "PoB and PoS," where Proof-of-Burn blocks act as checkpoints, and "PoW and PoS," where PoW blocks act as checkpoints containing no transactions, but anchor both to each other and to the PoS chain. Peercoin uses PoW/PoS consensus. To solve the "nothing at stake" issue, Peercoin uses centrally-broadcast checkpoints (signed under the developer's private key) according to which no blockchain reorganisation is allowed deeper than the last known checkpoints. Here the problem is that the developer becomes the central authority controlling the blockchain.

4.3. Gossiping—Blockchains are also decentralised, redundant storage systems. This redundancy makes it very difficult to hijack the information stored in them. How this information travels through the network of computers is a characteristic that varies from one blockchain system to another. Given the lack of a central routing authority (like it would exist, for example, in traditional electronic payment systems) nodes must transmit the information they possess—in general new blocks, but it may be also the full blockchain to new nodes that enter the network—to peers they know are participating in the system. To this aim, nodes possess a list of peer nodes. Whenever a new block is added to the local blockchain of a node, it passes the block to others in its peer list by *Gossiping*.

We identify two possible layouts for *Gossiping*:

(1) **Local.** *Gossiping* occurs first in a local manner (through a local validation process) until consensus is reached. This is also called "federated consensus" used *e.g.*, in Ripple, in which nodes can share transaction records to another node and reach consensus without directly knowing all the nodes in the network.[32] Therefore most information travels "locally"—in terms of the P2P network—such that a consensus is reached at this initial level. Only then is the information sent throughout all the other nodes. In this layout, the Gossiping can be termed "local."

(2) **Global.** In most implementations—Bitcoin, Ethereum, etc.—Gossiping occurs to a list of peers that have been selected by what in the Bitcoin network are called fallback nodes. These fallback nodes maintain a list of all peers in the network. Upon connection of a new node, they submit a randomly chosen list of peers to the entrant one. The logical network topology is intended to be largely unstructured, similar to the Erdos-Renyi network.[33] Such topology lacks a concept of vicinity or local neighbourhood, and therefore the Gossiping process can be termed *global*.

4.4. Consensus Agreement—A *consensus agreement* defines the set of rules under which records (like sets of transactions or any other atomic piece of information) are independently updated by the nodes of a distributed system. This is important to understand how a distributed system is able to handle the so-called Byzantine failures, *i.e.*, how the system composed of n nodes can achieve consensus on storing verified, trustworthy, information even in the presence of f malicious nodes or in presence of malicious participants launching sybil attacks.[34] In this

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

regard, it is very important to understand how the nodes communicate between them.

4.4.1 Latency

Latency is a sub-subcomponent which describes the rule of message propagation in the networks.

(1) **Synchronous Communication.** Systems which set upper bounds on "process speed interval" and "communication delay" such that every message arrives within a certain known, predefined, time-interval (Δ) are called *synchronous*. This does not preclude the possibility of having message delays due to exogenous network latency, but the delay is bounded and any message that takes longer than Δ is discarded. Lax synchronicity assumptions apply also to the Bitcoin blockchain. For example, a block is rejected if contains a timestamp: 1) lower than (or equal to) the median timestamp of the previous eleven blocks; and 2) greater than (or equal to) the "network-adjusted time" plus 2 hours. Another example of blockchain adopting synchronous communication is Ripple, through the use of clocks. Specifically, Ripple's "LastLedgerSequence" parameter asserts that a transaction is either validated or rejected within a matter of seconds.

(2) **Asynchronous Communication.** Systems which do not set any bound on "process speed interval" and "communication delay," such that every message/packet can take an indefinite time to arrive, are called *asynchronous*. Although this type of communication protocols brings some advantages (*e.g.*, calls/requests do not need to be addressed to active nodes and nodes do not need to be available when a new information is sent to them by peers), its main disadvantage is that response times are unpredictable and it is harder to design applications based on them. Synereo is an example of a blockchain using an asynchronous communication protocol.

4.4.2 Finality

Finality describes whether information intended to be stored in a blockchain (or, as a matter of fact, in any system) can be safely considered *perpetually* stored once the recording is performed. For a distributed system like blockchain-based ones this is very challenging to achieve, and it is certainly not one of the underlying design principles. In a system where the new blocks diffuse through gossiping, and because of rules such as the precedence of the longest chains, even if consensus is achieved globally, *a priori* nothing prevents a set of new nodes entering the system and overriding the previous consensus by offering a longer versions of the history. We identify two possible layouts for Finality:

(1) **Non-Deterministic**. In this case, the consensus agreement "eventually settles." Non-deterministic systems use randomised or inherently probabilistic consensus (also called *stabilising* consensus) in which the probability to disagree decreases over time. For example in the Bitcoin blockchain the block frequency is adjusted (with respect to the block-mining rate and indirectly to the computational power of the nodes) to minimise the probability of forks. Moreover, the propagation of blocks through the network has characteristic delays and even in the presence of only honest nodes the fork probability cannot be ruled out simply because different nodes may find competing blocks of the same height before the one found first reaches the complete network.[35] This cannot be prevented even if there is in place a concurrency control mechanism, which which

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

attempts to correct results for simultaneous operations. Therefore, overall, the protocol is non-deterministic. Thus, even though the widespread heuristic "wait until 6 confirmed blocks are appended to the chain" reduces the likelihood that a transaction is overridden afterwards, it does not eliminate completely the probability of a previously validated block being pruned and removed from the blockchain in the future.

(2) **Deterministic**. In this case, consensus agreement converges with certainty and transactions are immediately confirmed/rejected in/from the blockchain. This property turns out to be very useful for smart contracts where, using state-machine replication, consistent execution of the contracts can be achieved across multiple nodes. All blockchains based on Lamport Byzantine Fault Tolerance achieve deterministic consensus.[36] A prime example of an implementation featuring deterministic finality is Stellar. Another example of deterministic systems is in private blockchains, where new blocks follow a predefined set of rules.

5. Transaction Capabilities

The second main component, *Transaction Capabilities*, is important to illustrate scalability of transactions and usability in possible applications and platforms. One of the major challenges for blockchain technologies is to increase the transaction throughput to compete with other solutions already available in the market (*e.g.*, centralised payment systems, like credit cards). In order to achieve these improvements, quantitative parameters (*e.g.*, data storage in block header, transactions per second (or TPS)), need to be redesigned to realise such improvements. Figure 1 illustrates the subcomponents forming the component Transaction Capabilities:

- [1] Data Structure in the Blockheader
- [2] Transaction Model
- [3] Server Storage
- [4] Block Storage
- [5] Limits to Scalability
  - [5.1] Transactions
  - [5.2] Users
  - [5.3] Nodes
  - [5.4] Confirmation Time

5.1. Data Structure in the Blockheader—The data stored in the block header has different functions. On the one hand, it includes the transaction hashes for validation purposes; on the other, it contains additional information for different application layers or blockchain technology platforms. The *data structure in the blockheader* describes the capabilities of the system to store transaction information. The original application of Merkle proof was implemented in Bitcoin, as described in Nakamoto's white paper.[1] We identify two possible layouts for data structure in the blockheader:

(1) **Binary Merkle Tree.** Bitcoin uses the *binary Merkle tree* within the block header to store the transactions.[37] The information in the block header in the Merkle tree structure contains a hash of the previous header, timestamp, mining difficulty value, proof of work nonce, and root hash for the Merkle tree containing the transactions for that block, which are used for the verification process to scale up the transactions speed. By convention, the

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

longest chain (from the so-called Genesis block) is considered to be the current status of the blockchain.

(2) **Patricia Merkle Tree.** One the one hand, the *Patricia Merkle tree* or simply *Patricia tree* (from **P**ractical **A**lgorithm **T**o **R**etrieve **I**nformation **C**oded **I**n **A**lphanumeric)[38] allows activities like inserting, editing or deleting information referring to the balance and nonce of accounts, which enables faster and more flexible validation of transactions than the binary Merkle tree.[39] However, with respect to applications, it has the important advantage of allowing for verification of specific branches of the tree. Ethereum uses the Patricia tree within the block header to store more information than what is possible in the binary Merkle tree.[40] Those contain transactions, receipts (essentially, pieces of data showing the effect of each transaction), and state.[39] Importantly, this technology allows even blocks outside the longest chain to contribute to the validation process, building a confirmation system that is less centralised. This is the so called Ghost rule, a variant of which is also implemented in the Ethereum blockchain.[39]

5.2. Transaction Model—The *transaction model* can be imagined as an accounting ledger which tracks the inputs and outputs of each transaction. The transaction model describes how the nodes connected to the P2P network store and update the user information in the distributed ledger. The challenge of the transaction model is to prevent data that ought not to be trusted by the parties connected to the system—*e.g.*, data originating in untrustworthy behaviour, like double spending—from entering into the ledger

As of the time of writing, it is possible to identify two possible transaction model layouts in the more widely used blockchain-based systems:

(1) **The Unspent Transaction Output (UTXO).** The *UTXO* model includes a refractory number of blocks (currently 100 for Bitcoin) during which network participants are prevented from using the transaction output in new transactions. In this way, it forbids miners from spending transactions fees and block rewards before stable validation status of the block chain. This measure prevents the *forking problem* of blockchains. This transaction mechanism is available in blockchain technologies like Bitcoin.[41]

(2) **Traditional Ledger.** In comparison to the UTXO model, different implementations of blockchain systems—like Stellar and Ripple—use a more traditional ledger model to record the transactions recorded in the system. In particular, Stellar lists every single transaction in the Stellar distributed ledger history. Also, Ripple uses the traditional ledger transaction model to register increments/decrements of balance and clear all account balances. In Ethereum some transactions are used to execute actions in smart contracts defined in specific atomic records in the blockchain. Those transactions can be seen as order executions of stakeholders which perform the actions out of said smart contracts.

5.3. Server Storage—At the core of blockchain-based systems is their decentralised nature. This requires that nodes connected to the peer-to-peer network are indistiguishable from each other. This concept, however, cannot be fully realized when the storage needs, computing power, or bandwidth constraints of the network nodes do not permit it. In these scenarios, different nodes have access to different layers of information, and those which do not store the information fully are "thin clients" connected to the peer-to-peer network.[42] We identify two possible layouts for *Server Storage*:

(1) **Full Node Only.** All nodes connected to the network, and which are part of the validation

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

process, are of the same kind. This is a genuinely peer-to-peer network where all the nodes are equivalent in terms of information contained. This property creates a large degree of information redundancy, which makes the system more resilient to attacks or malfunctioning.

(2) **Thin Node Capable.** In this setup, some nodes connected to the network contain only a selected subset of all the information contained in the blockchain. This creates more scalable systems (in terms of number of nodes connected to the network and the concomitant network traffic and storage needs), but may deteriorate the resiliency as only a fraction of the nodes contain the complete blockchain information.

5.4. Block Storage—The information stored in the blockchain determines the scalability of the system across some dimensions. More crucially, it also allows us to understand how concomitant information from users are abstracted within the system. We identify two possible layouts for *Block Storage*:

(1) **Transactions.** In systems like Bitcoin, only the transactions are stored. They contain a set of inputs and outputs that help to identify the emitter(s) and receiver(s) of a specific transaction. This kind of approach is preserved in more exotic applications of blockchain-like technologies, like IOTA, which relies on the storage of a directed-acyclic-graph to store every single transaction. This kind of approach works not only for cryptocurrency applications, but also underlies all property-transfer-like applications.

(2) **User Balance.** In systems like Ripple, the decentralised storage also contains information about user balances. This approach may limit the storage needs of the system, but at the same time reduces accountability and the possibility to roll back transactions.

5.5. Limits to Scalability—The decentralised nature of blockchain systems and the concomitant redundancy in storage impose different kinds of limits to the way in which a specific implementation scales with *system size*. System size is used here in a broad sense: it may refer to the number of nodes connected to the network, the number of users of the service, the set of network connections and/or amount of network traffic, the number of transactions, etc. It is worth remarking that these ingredients are intertwined in the real world,[43] and—depending upon usage and continuing development—the limiting factor of a particular blockchain system may vary over time. Because blockchain technologies are rapidly evolving at present,[44] these limits are often changed in the course of platform development. An example of a change in scalability brought by technological evolution is the adoption of Segregated Witness (Segwit) technology.[45] This technology (under the name BIP141) provided a solution to the problem of transaction malleability and enabled the storage of more transactions per block. Although some exponents of the Bitcoin community criticized Segwit to be a non-sufficient short-term scaling solution,[46] it triggered interesting side consequences for the Bitcoin system: first, the implementation of micropayment channels where the transactions are not necessarily recorded in the blockchain but take place off-chain between multiple trusted parties;[47,48] and second, the proposal of SegWit2x, which would have enabled an alternative approach to store information in the transactions, modifying the scalability properties of the platform by increasing the block size to two megabytes.[49]

The importance of this component is due to its influence on the final scaling of the system. Scaling is a property that specifies how growth will influence its overall performance. As an example, how the total network traffic induced by unverified transactions grow with the number

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

of network nodes. If every node has a small—limited—number of connections, then the total network traffic will scale linearly with the number of nodes. In mathematical terms it will be $\mathcal{O}(N)$. However, if every node is connected to each other, then the traffic will be $\mathcal{O}(N^2)$, *i.e.*, it will grow quadratically with the number of nodes. Therefore, if—for a given implementation—network traffic is the most crucial limiting factor, different logical topologies will have different scalability. Acknowledging that a categorical definition is a crude simplification, we will focus here on the most limiting element for each system. We identify five possible layouts, categorized by their respective *Limits to Scalability*:

(1) **Number of Transactions.** The most common real-world example is found in Bitcoin, which has a limitation in the number of transactions it can process in every block, because of the hard-coded limit to the block size in bytes. Given that new blocks appear (on average) every ten minutes, this means that the number of transactions that can be included in a given time window is limited. Therefore the layout "Number of Transactions" refers—regardless of the information stored in the blockchain—to the specific implementations,[50] where the number of operations that can be included in the blockchain is severely limited by design.

(2) **Number of Users.** Bitcoin only stores transactions in its public ledger. This is different from other related technologies. Ripple, on the other hand, stores not only transactions, but also the state of the Ripple accounts. Therefore, in scenarios like this, it is the number of users of the system that limits its scalability. A similar problem occurs in Ethereum where the system will be constrained by the number of DAOs, individuals, etc. that it will contain, as these are the actors that generate activity in the system. Therefore, the term "Number of Users" for this layout is a broad reference to the number of objects with stored states. Needless to say, this layout is somewhat linked to the first, in that the number of transactions will depend on the number of users. However, each limiting factor can still appear irrespective of the other.

(3) **Number of Nodes.** The number of nodes connected to the network, acting as verifiers for the information that is stored on the blockchain, presupposes a limiting factor because of the mechanism of information diffusion adopted. Gossiping is a process that requires larger times in decentralised networks to propagate into a consensus state,[51] and may even reach a point—when the relative time taken by network traffic is very long—where consensus cannot longer be reached and the blockchain naturally forks. Therefore this process naturally limits the applicability of fully decentralised solutions.

(4) **Possible Values.** Each of these three layouts can have different values, depending on how detrimental a specific layout is to the overall performance of the system. The possible values that each layout can have are divided into four: (i) **indifferent**, (ii) **at most linear**, (iii) **at most quadratic**, and (iv) **worse than quadratic**. The first value describes when the relevant global characteristics of a system are independent of the number of specific classes;[52] the other three categorical values depend on the number of elements in said class. For example, the number of users is largely irrelevant to the performance of the Bitcoin network (because this number is never translated into any property of the network). However, the number of transactions is linearly related to local network traffic.

(5) **Confirmation Time.** The time it takes a specific action to be confirmed ultimately depends on the time it takes for it to be added to the blockchain, and to be validated by

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

further blocks later appended to it. Different approaches can be taken to this process: **deterministic** addition of new blocks at regular intervals (taken by Peercoin) and **stochastic** addition like in Bitcoin, where the process of mining induces an exponential distribution of inter-block discovery time.

6. Native Currency/Tokenisation

So far, cryptocurrencies and other transfer of property records are the most common usage of blockchain technologies. In cryptocurrencies, system participants who contribute to the verification process—if selected by some rule to issue a new block into the blockchain—are awarded the possibility to issue a transaction without issuer (so called "coinbase") to themselves. On the one hand, this is a customary way of introducing new assets into the system. On the other, it introduces an incentive for users to participate of the verification process which leads to an increased trustworthiness on the system.

The aforementioned incentive scheme is to be provided in a token,[53] whose value is assigned precisely because of the cost associated with its production.[52,54] Initially, solutions like Bitcoin created their own (and single) asset class (*the bitcoin*) that can be used for transactions within the system. This particular solution is not the only one possible—for example Ethereum, where beyond the native token, Ether, arbitrary new tokens can be created and exchanged via smart contracts. Further, the native currency possibilities (present, for example, in Ripple and tokenisation) enable different use cases of blockchain technologies like asset-transfers via tokens, exchanges, etc.[55] All this is just the beginning of cryptoeconomics: it is of the utmost importance how these assets are supplied to the system, because this affects the way users are incentivised to participate in the validation process. Figure 1 (above, page 107) illustrates the subcomponents forming the component Native Currency/Tokenisation:

- [1] Native Asset
- [2] Tokenisation
- [3] Asset Supply Management

6.1. Native Asset—Some systems implemented using blockchain technologies have an underlying native asset (which are normally called *cryptocurrencies*) which are digital tokens whose owners assign a value and which allow for the daily activities on the platforms or in their respective communities. Opinions may vary on whether these cryptocurrenties ought to be considered fiat or commodity currencies,[56–58] and whether they may eventually be massively adopted replacing traditional ones.[59] We identify three possible layouts for *native assets*:

(1) **None.** Private blockchain implementations do not require a native asset within to incentivise participation. In these cases, there is no native asset incorporated into the system

(2) **Own Cryptocurrency.** Most implementations of cryptocurrencies only deal with transfer of property of its own tokens within the system. Bitcoin and Litecoin are examples of technologies with single asset compatibility.[60] These technologies are limited to their own underlying digital currency, but can also have off-chain solutions to interoperate with other currencies, in order to execute transactions or to make use of smart contracts. There are also solutions like coloured-coins.[61]

(3) **Convertible Multiple Assets.** Other technologies like Counterparty and Ardor do have

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

their own underlying currencies or tokens to execute tasks. However, these technologies also enable the possibility of exchange of assets expressed in others beside those native to the platform. This approach of multiple, convertible currencies has the advantage of allowing for exchange markets be directly incorporated into the system.

6.2. Tokenisation—A token acts as a digital bearer bond, whose ownership is determined by the data embedded in the blockchain. Ownership of the tokens is transferable between holders using other transactions with associated "transfer" metadata. This does not require the approval of any other authority. The possibility of tokenisation enables a range of possible use cases for blockchain technologies outside the purely financial world.[52,55,62–64]

We identify three possible layouts for *Tokenisation*:

(1) **No Tokenisation Present.** Bitcoin does not have native technologies that enable tokenisation and requires third-party technologies.
(2) **Tokenisation Through Third-Party Add-Ons.** Bitcoin plus Colour-Coin enables the existence of tokenised transactions in the Bitcoin blockchain.[61] Said solution is based on the cryptographic nature of Bitcoin addresses and the script language.
(3) **Tokenisation.** The tokenisation possibilities together with the extensions of metadata are available in several implementations and constitute the backbone of blockchain-based property registries. The most paradigmatic example is Ethereum, where the creation of a new token is produced by means of the creation of a smart contract. Thanks to this flexibility, and the extreme possibilities for the expansion of such a platform, the conditions for creation of new tokens are countless.

6.3. Asset Supply Management—The process of digital asset (*i.e.*, cryptocurrency) creation varies across different blockchain technologies. Each approach has taken different economic frameworks, in most cases fixing a specific monetary policy for the future of a particular system. It is also generally a pillar of the incentive scheme that users have to participate (or not) in the validation process.[51]

We identify three possible layouts for *Asset Supply Management*:

(1) **Limited-Deterministic.** The most replicated system in the world of blockchain is the limited supply as introduced in Bitcoin. Not only does the supply grow sub-linearly over long periods of time (in contrast to what occurs in normal fiat currencies), but it is designed to have a well-defined limit. It is important that, while this incentivises users to adopt the technology and contribute to the process of verification—for which they receive remuneration—on the other hand, it also creates an incentive to hoard the asset, limiting transactions.
(2) **Unlimited-Deterministic.** Very few blockchain-based digital currencies have attempted to create an unlimited supply, and those that have (for example Dogecoin or Freicoin) have not seen wide acceptance.
(3) **Pre-Mined.** Some altcoins (with the purpose of funding the development of a platform, or with the sole idea of profiting) have distributed all the assets before the starting of the system. From that point on, a reward system induces some kind of redistribution.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

7. Extensibility

The alignment of the interoperability, intraoperability, governance, and script language determine the future ecosystem of the blockchain network and the integration possibilities of a variety of blockchain related technologies. Figure 1 (see above, page 107) illustrates the subcomponents forming the component Extensibility:

- [1] Interoperability
- [2] Intraoperability
- [3] Governance
- [4] Script Language

7.1. Interoperability—Interoperability illustrates the overall capability of blockchains to exchange information with other systems, outside of blockchains. It allows inflow, outflow, and information retrieval of data providers that are not necessarily a blockchain-based system, *e.g.*, financial data providers.[65] We identify three possible layouts for Interoperability:

(1) **Implicit Interoperability.** This occurs when the smart contracts that specify conditions under which a particular transaction (or event) is to take place can be written in a Turing-complete blockchain script language. In this context, implicitly any kind of condition can be specified, even those involving specific statuses in other systems. This implies an (albeit cumbersome) way of interaction between a blockchain solution and any API tool or interface.

(2) **Explicit Interoperability.** If the script language is not Turing complete or the system has specific tools implemented that enable interoperability with the real world (like Bitcoin with Counterparty), then we term this explicit interoperability, as it is brought purportedly into the system as one of its design principles.

(3) **No Interoperability.** A blockchain without any possibility for interacting with other systems. As implemented, Bitcoin in the absence of external solutions (*i.e.*, off-chain layers) has no interoperability implemented. It applies to most existing blockchain-based systems whose script language is not Turing complete.

7.2. Intraoperability—Intraoperability illustrates the overall capability of blockchains to exchange information with other blockchains. It allows inflow, outflow, and exchange of data between different blockchains.[66] We identify three possible layouts for *Intraoperability*:

(1) **Implicit Intraoperability.** This occurs when the smart contracts that specify conditions under which a particular transaction (or event) is to take place can be written in a Turing-complete blockchain script language. In this context, implicitly any kind of condition can be specified, even those involving specific statuses in other blockchains.

(2) **Explicit Intraoperability.** If the script language is not Turing complete but is specifically designed to allow for intraoperability, then we term this explicit intraoperability, because it is brought purportedly into the blockchain and is one of its design principles. An example of this is Bitcoin with Counterparty.

(3) **No Intraoperability.** A blockchain without any possibility for interaction with other blockchains. As implemented, Bitcoin in the absence of external solutions has no intraoperability implemented. Solutions for non-intraoperable blockchains rely on: 1) Trusted proxies to connect blockchains; 2) Pegged blockchain systems; 3) Distinguishing tokens within the same blockchain based system.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

7.3. Governance—Effective governance rules are crucial for the successful implementation of blockchains and for their capability to adapt, change and interact. As blockchain deployment structures (public chain, private chain, consortium chain) are different, their management patterns are also quite different. We identify two types of governance rules: 1) *technical rules* of self-governance, defined by the participants. Technical rules are composed of software, protocols, procedures, algorithms, supporting facilities and other technical elements; 2) *regulatory rules*, defined by external regulatory bodies composed of regulatory frameworks, provisions, industry policies, and other components.[67–69] Regulatory rules are by definition not technical in nature and therefore outside the scope of this taxonomy. We focus instead on technical rules which are particularly interesting for the way they feed back into the proposed technological solutions. We identify three possible layouts for *Governance*:

- **Open-Source Community.** In this case, open communities of protocol developers (following open-source principles) and validators (very often in coordination with the blockchain foundation) coordinate upgrades and technical adjustments of the blockchain. For example, Bitcoin is mainly maintained by a team of core developers who in coordination with miners agree on changing parameters or other settings of the Bitcoin network. Ethereum and Hyperledger (backed up by the Linux Foundation) also follow an open-source community model. Open-sources governance has been criticized for turning out to become sometimes too centralized and obstructive.[51,70] For example, the Bitcoin protocol core developers can veto external proposals to change rules and big mining pools can signaling support for an upgrade earlier than expected.
- **Technical**. Since blockchain technologies are very versatile and can be applied to many business cases, enterprises with a measure of technical strength (*e.g.*, IBM and Microsoft) have proposed themselves as technical solution providers for blockchain architectures (*i.e.*, providers of proprietary hardware and software systems and basic services). In these cases, the technical rules of blockchain governance are dictated by the companies according to their business goals. For example, in 2015 Microsoft collaborated with ConsenSys to create an Ethereum blockchain technology service and implemented it as part of the Microsoft Azure service (EBaaS) to provide distributed ledger technology trials for enterprise customers, partners and developers. In order to protect their often proprietary blockchain architectures, these companies generally apply also for patents. According to a 2016 World Economic Forum report, 2,500 patents on this topic have been filed from 2014 to 2016.[2]
- **Alliance**. This is the blockchain governance model proposed by industry consortia (*e.g.*, B3i, R3) composed of companies with common business or technological progress demands. The alliance model has the scope to create shared technology platforms and to build common business models and standards. Only companies that meet certain criteria (*e.g.*, payment of the fees, qualification of the organisation) are given license to collaborate to set the technical rules of blockchain governance. Those companies join together to promote commercial and technological progress in the area of blockchain technology to their mutual benefit based on common contribution.

7.4. Scripting Language—Widespread programming languages are Turing-complete, which in formal terms refers to the fact that it is possible to implement an algorithm on it to simulate any Turing machine. These are therefore general purpose languages, in which arbitrary computations

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

can be performed. Languages that are not of this kind, are so because of design reasons which aim to prevent specific behaviours of code execution, like undefined termination.

Blockchain systems make it possible to modify the conditions under which certain information (*e.g.*, transactions) will be included in the public record. These conditions must be specified in an algorithmic manner, and in some contexts are termed *smart contracts*. These algorithms are elicited in a *scripting language* designed purely for this purpose. Therefore the intended flexibility given to the users, with respect to the scope that the algorithm can develop, affects tremendously the degree of freedom to create conditions for some actions to occur (on the one hand) and the hypothetical computational effort that may be necessary to assess if a particular condition is fulfilled or not.[71]

It is worth remarking here that the degree to which the scope of the scripting language is limited is another design decision developers must carefully consider before the implementation of the blockchain, as abrupt changes (or bugs) may undermine the logic of particular transactions. We identify four possible layouts for Scripting Language:

(1) **Turing Complete.** Ethereum refers to a suite of protocols that define a platform for decentralised applications. With respect to scripting languages, on the one end of the spectrum, the Ethereum Virtual Machine (EVM) can execute code of arbitrary algorithmic complexity. In the terms described above, Ethereum is "Turing complete," because developers can create applications that run on the EVM. Furthermore, Counterparty also uses Ethereum's entire smart contract platform to enable users to write Turing complete smart contracts. It has been pointed out that there exist scalability and security concerns regarding the usage of Turing complete scripting languages in blockchain systems.[72] As of this writing, these have not been resolved.

(2) **Generic Non-Turing Complete**. When designing Bitcoin, a decision was made to keep the scripting language limited in scope, to reduce the impact of these calculations on the efficiency of the system. It is therefore a non-Turing complete language, and most blockchain implementations have followed this path. There is no connectivity in these to so-called "oracles" that allows data to be obtained from sources that gather data that is exogenous to the blockchain.

(3) **Application-Specific Non-Turing Complete**. There are some non-Turing complete languages that are more expressive than the generic ones and purposely designed for certain cases. By restricting the language to be only able to write programs relevant to specific limited cases, the potential outputs of those programs become predictable. This allows those outputs to be queried and easily analysed. One example is Digital Asset Modelling Language (DAML) which is designed to codify only financial rights and obligations for execution in private networks. DAML is also more expressive than Bitcoin's scripting language and easier to read for a non-technical audience.

(4) **Non-Turing Complete + External Data.** There exists a third category, barely used so far, that (while keeping the nature of the scripting language non-Turing complete) allows for the existence of oracles. These oracles are considered trustful sources and add a layer of simplification to the validation to be performed by the language, empowering above Turing-completeness (as long as the oracles are reliable). This layout is thereby deemed "non-Turing Complete + External Data."

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

8. Security and Privacy

The recent evolution and new implementations of blockchain systems bring risks, both technical and operational, associated with security and privacy. Thus, we group together security and privacy as two interrelated faces of the same problem. Similarly, ISO TC 307 has created a dedicated Working Group on "Security and Privacy."[73]

Security of blockchain systems is a matter of significant concern. Cryptocurrencies, the most widely deployed application of blockchain systems, have suffered from cyberattacks which became possible because of the mismanagement of sensitive data and the flawed design of the systems.[74] Without going into the detailed distinction between "risks," "threats," "attack surfaces," and "vulnerabilities," the security of blockchain systems generally comprises: 1) Information mismanagement (alternation, deletion, destruction, disclosure, etc.); 2) Implementation vulnerabilities (including cryptomechanisms' implementation vulnerabilities, run-time leakage of information, etc.); 3) Cryptographic mechanism mismanagement (including use of weak algorithms, key disclosure, etc.); 4) User privilege mismanagement. For a recent comprehensive survey specifically detailing the security and privacy aspects of Bitcoin and its related concepts, we refer the readers to Conti (*et al.*) "A Survey on Security and Privacy Issues of Bitcoin" (2017).[75] When we speak of privacy we are referring to "freedom from intrusion into the private life or affairs of an individual when that intrusion results from undue or illegal gathering and use of data about that individual," as cited in ISO 25237 and other ISO standards. These privacy principles apply to any ICT system containing or processing PII, including blockchain systems. Figure 1 (above, page 107) illustrates the subcomponents forming the component Security & Privacy:

- [1] Data Encryption
- [2] Data Privacy

8.1. Data Encryption—By *Data Encryption* we refer to cryptographic primitives. Cryptographic primitives (cryptographic algorithms) are used to ensure authenticity, integrity, and order of events. For example, Bitcoin blockchain uses the ECDSA digital signature scheme for authenticity and integrity, and the SHA-2 hash function for integrity and order of events. Hash functions are also commonly used as a part of a Proof-of-Work consensus mechanism. We identify two major layouts for Data Encryption:

(1) **SHA-2.** SHA stands for Secure Hash Algorithm. In its two incarnations, SHA-256 and SHA-512, SHA (originally developed by the National Security Agency of the United States of America) is the most widely-used for hashing functions, likely stemming from its use in Bitcoin.[76,77] When applied to hash transactions, it requires a piece of information from the issuer, *i.e.*, the public key for the validation to take place.[78]

(2) **ZK-SNARKS.** The Zero-Knowledge—Succinct Non-interactive Argument of Knowledge is a newer technology where no data has to be provided to validate a specific hash,[79] but the hash itself serves as proof. Therefore both the hashed message and the encrypted one are sufficient proof of existence. This novel approach anonymises the individual information to a much greater degree.

8.2. Data Privacy—Although public/private key infrastructures and other measures like hashing functions should ensure that only the intended recipient can read the message and have

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

access to the content of the transaction, the research shows that blockchain transactions (*e.g.*, in Bitcoin) can be linked together in order to extract additional information and eventually also the identity of the participants.[80] Indeed, there exists an inevitable tradeoff between a decentralised peer-validated system and the security and privacy of information. In this regard, several alternative solutions have been proposed to "encrypt" the data in such a way that even though computations and transactions occur in plain sight, the underlying information is kept completely obfuscated. Obfuscation is a way of turning any program into a "black box." This is equivalent to the original program: it runs the same "internal logic" and provides the same outputs for the same inputs, but information on the data and processes is inaccessible. Of course there exists a strong interrelation between *Data Privacy* and *Data Encryption*. Based on the solutions proposed so far to enhance Data Privacy, we identify two possible layouts:

(1) **Built-In Data Privacy**. With built-in data privacy we include all those blockchains that by default provide obfuscation of information. For example ZeroCash uses built-in zero-knowledge cryptography to encrypt the payment information in the transactions.[81] Although ZeroCash payments are published on a public blockchain, the sender, recipient, and amount of a transaction remain private. Alternatively, blockchains like Enigma (a project that seeks to implement the "secret sharing" DAO concept) uses built-in secure multi-party computation guaranteed by a verifiable secret sharing scheme.[82] In this case, the data can be split among N parties in such a way that $M < N$ are needed to cooperate in order to either complete the computation or reveal any internal data in the program or the state. But M-1 parties cannot recover any information at all (which implies the need for trust on the majority of the participants to be honest). Finally, CORDA by R3 proposes a Node to Node (N-to-N) system characterised by encrypted transactions where only the parties involved in the transaction have access to the data.[83] This is suitable for financial transactions where a high degree of confidentiality is required. Third parties like central banks or other market authorities may have access to the data by invitation only.

(2) **Add-On Data Privacy**. In this case, pseudonymous or public blockchains must resort to external solutions in order to obfuscate the information. One method is the *mixing* service like Coinjoin. The principle behind this method is quite simple: several transactions are grouped together so to become a unique M-to-N transaction. If for example, Alice wants to send one coin to Bob, and Carla wants to send one coin to David, a mixing transaction could be established whereby the addresses of Alice and Carla are both listed as inputs, and the addresses of Bob and David are listed as outputs in one unique transaction. Thus, when inspecting the 2-to-2 transaction from outside it is impossible to discern who is the sender and who the recipient.[84,85] Alternative to mixing services, *secret sharing* allows data to be stored in a decentralised way across N parties such that any K parties can work together to reconstruct the data, but K-1 parties cannot recover any information at all. Alternative add-on data privacy tools include *ring signatures* and *stealth addesses* which hide the recipient of a transaction and can be used by any blockchain.[86,87] Ring signatures—firstly introduced by Rivest, Shamir, and Tauman in "How to Leak a Secret" (2001)[88]—and its variant, linkable ring signatures, allow users to hide transactions within a set of others' transactions. In this case the transaction is tied to multiple senders' private keys but only one of them is the initiator. Thus, the verifier may only identify that one of them was a signer, but not who exactly that was. In the case of stealth addresses, a

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

receiver generates a new dedicated address and a "secret key" and then sends this address to someone who he wants payment from. The sender use the address generated by the receiver plus a "nonce" (one time random number) in order to generate the address he/she will send funds to. The sender communicates the nonce to the receiver who can unlock the address by using the nonce and the secret key generated earlier. Monero is an example of a blockchain that aims to achieve privacy through the use of traceable ring signatures and stealth addresses.

9. Codebase

The codebase of blockchain technologies delivers information about which challenges a developer could face and what kind of changes the underlying programming language could undergo. Therefore the main component 'Codebase' is essential to align and increase the efficiency of blockchain-related IT architectures. Figure 1 (above, page 107) illustrates the subcomponents forming the component Codebase:

- [1] Coding Language
- [2] Code License
- [3] Software Architecture

9.1. Coding Language—Coding language illustrates the interconnectivity of programming languages of blockchain technologies. We identify two possible layouts for *Coding Language*:

(1) **Single Language.** Bitcoin has released Bitcoin Core version 0.13.1 in the underlying coding language C++. As Bitcoin is open source, implementations occurred (much less popular than the original codebase) in different languages (like Java).

(2) **Multiple Languages.** Ethereum uses C++, Ethereum Virtual Machine Language, and Go, which enables more interaction with other languages. Stellar maintains JavaScript, Java, and Go-based SDKs for communicating with Horizon. There are also community-maintained SDKs for Ruby, Python, and C-Sharp.

9.2. Code License—The Code License illustrates the possibility of changes to the source code of the underlying technology. We identify thee possible layouts for *Code License*:

(1) **Open-Source.** Regardless of the exact licence used for specific projects, we refer only to the openness in the source code as the only differentiating factor. Bitcoin core developers have continuously licenced the source code under the MIT licence. Counter-intuitively, a permissive licence like the MIT one (in which other developers can take the source code and fork it) eventually prevents multiple implementations. It also allows for continued development, more code growth, and adoption at a faster pace. Ripple and Stellar have licensed their codes with the ISC License, which is another permissive licence.

(2) **Closed-Source.** For private implementations of blockchain-based systems, the source code is not necessarily openly distributed. Just as an example, most the blockchains running on the Ethereum Enterprise Alliance (rather than on the public Ethereum blockchain) use closed-source codes. In this case, risking the existence of unadressed bugs or unreported characteristics that may violate the expected conditions of use and functioning, the code may be kept outside of reach for users.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

9.3. Software Architecture—The *Software Architecture* refers to the high level structures of the blockchain system. Each structure is composed of software elements, the relations between them, and the properties that emerge from those elements and relations. The choice of the software architecture is very important in order to better manage changes once implemented. Software architecture choices include specific structural options among the possibilities that are available for software design.

We identify two possible layouts for *Software Architecture*:

(1) **Monolithic Design**. In this case, all the aspects of a decentralised ledger (P2P connectivity, the "mempool" broadcasting of transactions, criterion for consensus on the most recent block, account balances, nature of smart contracts, user-level permissions, etc.) are handled by a blockchain built as a single-tier software application without modularity. Examples of blockchains with monolithic design include Bitcoin and Ethereum. These architectures suffer from a lack of extensibility on the long run.

(2) **Polylithic Design**. The Polylithic approach decouples the consensus engine and P2P layers from the details of the application state of the particular blockchain application. For example, in Tendermint the blockchain design is "decomposed." Between the application process and its application-agnostic "consensus engine" (TenderminCore), it offers a very simple API which enables it to run Byzantine fault-tolerant applications, written in any programming language, not just the one the consensus engine is written in. Also Hyperledger Fabric follows a polylithic design as it is composed of interchangeable modules representing different components of blockchain technologies.

10. Identity Management

The main component *Identity Management* ensures secure access to sensitive data to establish a suitable governance model for the blockchain. This is a complex matter, as different levels of authority, accountability and responsibility are attached to different type of participants (*e.g.*, users, administrators, developers, validators, etc). Generally, the set of rules are defined and enforced through mechanisms intrinsic to the system itself (on-chain governance). The subcomponents also eventually determine the concept of digital identity that users end up having within the systems. Figure 1 (above, page 107) illustrates the subcomponents forming the component Identity Management:

- [1] Access and Control Layer
- [2] Identity Layer

10.1. Access and Control Layer—When establishing the right governance structure for a blockchain it is important to consider the ledger construction. Depending on its purpose, the ledger could be run by a central authority and governed by it, or it could be run in a decentralised fashion according to a set of governance rules adhered to and enforced by participants on the blockchain network. The governance structure determines the authorisation and the control policy management functions. Those rules provide permission for users to access to or use blockchain resources. Those are a set of rules that manage user, system and node permissions that must be followed in security-related activities. Blockchains may have different permissions according to different levels of access to and control of data. The distinguishing features must answer to the following questions:

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

- Which users have "read" access?
- Which users have "write" access?
- Is it there anyone who can "manage consensus" (*i.e.*, update and maintain the integrity of the ledger)?

According to the set of governance rules, we may have different system designs that reply to the above questions in a different way in order to better serve either a public or a private interest of either a *general* (like in the case of Ethereum) or a *special* (like in the case of Corda) purpose. On one side, private blockchains are generally those with a set of constrained "read/write" access alongside a consensus algorithm which allows only a pre-selected group of people to contribute and maintain blockchain integrity. Instead, public blockchains do not control "read/write" access or in the consensus algorithm for any given set of participants. Nevertheless, this does not mean that certain permission structures can not be implemented as part of a specific application.

Although different variations are possible, the authority to perform transactions on a blockchain generally belongs to one of the following main models of the *Access and Control Layer*:[89]

(1) **Public Blockchain**. In this case, there is no preference in access or in managing consensus. All participants (nodes), have "read/write" access and without any control can contribute to the update and management of the ledger. An example of a blockchain in this model is Bitcoin, where every participant can either choose just to use the blockchain to exchange Bitcoins (or other data on the top of it, in general by means of third-party technologies), run a full node, or even become a miner to participate in the process of transaction validation.

(2) **Permissioned Public Blockchain**. In this case "read" access is enabled for all users, however "write" access and/or "consensus management" require permission by a pre-selected set of nodes. Ripple belongs to this group, as to validate transactions a participant need to be part of the so-called *Unique Node List*. Some other examples include Ethereum and Hyperledger Fabric which is used for the exchange of tangible assets (real estate and hardware) and intangible assets (contracts and intellectual property) between enterprises.

(3) **Permissioned Private Blockchain**. In this case, "read/write" and "consensus management" rights can only be granted by a centralised organisation. One example of this type is Monax (formerly known as Eris).

10.2. Identity Layer—The onboarding and offboarding of nodes / entities to a blockchain network is handled differently by various software solutions. By *identification* we mean the capability to identify an entity uniquely in a given context. Digital identity can be defined as a set of identifying attributes of an entity that together enable the unique identification of the entity in a context (UID). A vital part of any identity system (and most information systems) is that a UID is managed throughout the entity's lifecycle to protect it from negligence and fraud, and to preserve the UID's uniqueness. A UID can then be assigned to the identity and used to link or bind the entity to the claimed identity and to any digital credential (software or hardware) issued to the entity. This digital credential acts a trusted proxy for the physical or logical entity and is used to support a wide range of personal and trust-related functions such as authentication, encryption, digital signatures, application logins and physical access control.

AML and KYC procedures—generally required to process personal or private data *e.g.*, medical data, bank information, or other personal data—are the key aspects to consider when

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

looking at the *Identity Layer*. We identify two possible layouts:

(1) **KYC/AML.**
Know-Your-Customer (KYC) and Anti-Money-Laundering (AML)-compliant blockchains have the ability to validate organisations and their attribute data from authoritative sources to ensure the quality of data written to the blockchain and linked to identifiers in the blockchain. One example is Stellar, which sets requirements for all integrators to implement KYC/AML identity verification processes to increase the transparency of the Stellar network participants. For another example, Ripple forces its financial services partners to implement an identity layer to verify the user information. The financial services partners have to perform due diligence, depending on the requirements they must fulfill.

(2) **Anonymous.** A common misunderstanding regarding the level of anonymity within Bitcoin networks is that the majority of the users do not distinguish between anonymity and pseudonymity. The Bitcoin protocol has no identity layer to identify the users, which could lead to misuse of Bitcoins and money laundering activities through its blockchain network, but to control approaches to anonymity in Bitcoin and other cryptocurrencies.[90] Regal Reid and Martin Harrigan have been able to demonstrate that several pseudonymous addresses can be linked to one single user.[91] See also Tasca, Liu, and Hayes, "The Evolution of the Bitcoin Economy" (2016) for a Bitcoin transaction-path driven user-identification method.[80]

11. Charging and Rewarding System

Blockchain systems incur operational and maintenance costs that are generally absorbed by the participants in the network. Different kinds of cost models are applied according to: 1) the architectural configuration design; 2) the governance system; and 3) the data structure and the computation required on-chain. One of the costs common to the wide majority of blockchains is the verification cost. This is required to sustain the validation process of the transactions that compete to be appended to (and never removed from) the ledger. The potential financial costs incurred when taking part in a blockchain platform require an incentive scheme that maintains consistency in the cost structure across the different stakeholders. Figure 1 (above, page 107) illustrates the subcomponents of the component we label the Charging and Rewarding System:

- [1] Reward System
- [2] Fee System
  - [2.1] Fee Reward
  - [2.2] Fee Structure

11.1. Reward System—This subcomponent illustrates the reward mechanisms automatically put in place and triggered by the systems in order to compensate active members contributing to data storage or transaction validation and verification. We identify two possible layouts for *Reward System*:

(1) **Lump-Sum Reward.** Individuals taking part in the storage, validation, and/or verification processes (*e.g.*, in Bitcoin verification is only performed by users called *miners*) may be rewarded for their action. For example, in Bitcoin, the first transaction in each block is called the *coinbase*, and the recipient is the user (or users) who created the block. The

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

lump-sum reward can be fixed, like in Enigma, or variable, like in Bitcoin.

(2) **Block + Security Reward.** In other blockchain-based technologies, like Ethereum, the blockchain rewarding system includes, besides the block reward, a reward for including forked blocks that are still valid in the validation. The purpose of the design is to incentivise cross-validation of transactions (crucial in a setting where validation can be arbitrarily costly).[92]

11.2. Fee System—A different kind of reward is that provided directly by the users to other participants in the system when launching any request in the network for storage, data retrieval, or computation and validation. With regards to this, we identify two sub-subcomponents: *Fee Reward* and *Fee Structure*.

11.2.1 Fee Reward

A *Fee Reward* is composed of the fees that users are required to contribute when using a blockchain. The fee system has been shown to play an important role in the way verifiers behave.[93,94] This kind of design-time consideration ought not to be neglected, though it often is. We identify three possible layouts for *Fee Reward*:

(1) **Optional Fees.** In Bitcoin and related technologies users can optionally pay a voluntary fee for the validation process.[95] This fee is optional, but it is assumed that the larger the fee is, the lower the processing time for a transaction (*i.e.*, the less time it will take for the transaction to be added to a block), as miners will be more incentivised to include it in a block. Moreover, given that the coinbase reward halves approximately every four years, currently, the reference Bitcoin client refuses to relay transactions with a zero transaction fee value.

(2) **Mandatory Fees.** Some systems like Stellar force all users to include fees in any transaction added into the system.

(3) **No Fees.** In comparison, Hyperledger Fabric is a blockchain solution for businesses, which combines a permissioned network and an identity layer without any transaction fees.

11.2.2 Fee Structure

When provided by the system, fees can follow either a fixed or a variable structure. There are two alternative layouts for *Fee Structure*:

(1) **Variable Fees.** In this case, the fee is somehow linked to the "size" of the request. In Bitcoin, the larger the transaction size, the higher the fee the user will pay in order to compensate for taking up space inside the block. Miners usually include transactions with the highest fee-per-byte value first. The user can decide how many "Satoshis" to pay per byte of transaction (a Satoshi is 0.00000001 Bitcoins, or one hundred-millionth of a Bitcoin). For example, if the transaction is 1,000 bytes and the user pay a fee of 300,000 Satoshis, he/she will be in the 300 Satoshi/bytes section (300,000/1,000=300.00). At the time of writing, this implies that the transaction will be included in the next 2 blocks (*i.e.*, within about 20 minutes). However, to avoid queuing, the user can opt to increase the fee. At the time of writing, the fastest and cheapest transaction fee is currently 360

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

Satoshis/byte. For an average transaction size of 226 bytes, a fee of 81,360 Satoshis is currently the cheaper fee in order to get the transaction included in the first available block without delays. Other blockchains apply variable fees and follow similar rules to Bitcoin.

(2) **Fixed Fees.** In this case, the fee is linked to the request, not to its "size." For example, in Enigma every request in the network for storage, data retrieval, or computation has a fixed price, similar to the concept of Gas in Ethereum. However, since Enigma is a Turing-complete system, the fee can be different depending on the specific request. Another example of a blockchain with a fixed transaction fee is Peercoin which requires a fixed 0.01 PPC per kilobyte.

12. Conclusion

In the 21st century, blockchain technologies will likely affect most, if not all, business areas: financial services,[96–99] IoT,[100,101] consumer electronics,[102] insurance,[103,104] the energy industry, logistics,[105,106] transportation, media,[107] communications,[108,109] entertainment, healthcare,[110] automation, and even robotics will be involved. After the creation of the Internet, it represents perhaps the most prominent technological innovation, and it will shape the forthcoming products and services in nearly every industry. Since the introduction of Bitcoin in 2009, the awareness of blockchain technologies has considerably increased. The first mover regarding the adoption of blockchain was the financial industry. This is explained by the fact that blockchain technology enables cost reductions and increases to efficiency in several business processes (both internal and external) for financial institutions. An example of the outsized impact of blockchain in the financial industry can be seen in the networks of global payments: monetary transactions in exchange for goods, services, or legal obligations between both individuals and economic entities. Beyond payments, blockchains allow for real-time settlements, which reduce operational costs for banks. Further, the immutability of the blockchain reduces the risk of fraud, and banks can use sophisticated smart contracts to capture digital obligations and to eliminate operational errors. Global payments are just a fraction of the overall use cases in the financial industry. In addition to this, many other industries, including the public sector, are now looking at blockchain-enabled solutions for their own processes. This trend is causing a proliferation of multiple blockchain architectures which often are not interoperable and are built according to different engineering designs. Lately, software architectures, companies, and regulators have begun to realise the need for standardisation of some of their components. This is becoming a necessary step for blockchain technologies in order to: 1) gain global adoption and compatibility, 2) create cross-industry solutions, and 3) provide cost-effective solutions. As always with standardisation processes, their creation must find equilibrium between different parties. But in this particular case, the open-source community that brought forward this disruptive technology—and which continues to develop the most implementations—should play a crucial role, as heralds of the advancement of blockchains and blockchain technologies.

Based on the review of the current literature on blockchain technologies, our work is an early stage analysis across existing software architectures with the aim of proposing a taxonomy: a reference architectural model for blockchains and their possible configurations. Organized on the basis of component-based design, the blockchain taxonomy deconstructs the various

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

blockchains into individual functional or logical components and identifies possible different layouts. It is hoped that this blockchain taxonomy will assist in the exploration of design domains, in the implementation, deployment, and measurement of the performance of different blockchain architectures.

Our work sheds light on the current proliferation of non-interoperable blockchain platforms and on the need for—and current discussions about—blockchain standards. Although our work contributes to the ongoing efforts to set blockchain standards, we do not conclude by insisting on the need for a set of standards *now*. This process generally takes several years in order to produce concrete solutions (even 10 years for complex subjects). Therefore, we think that our taxonomy represents a timely, honest intellectual exercise to be used as preliminary supporting material for all those interested in reducing blockchain complexity. At the same time, we are aware that our taxonomy tree, although hopefully very useful, is very preliminary and likely the first version of subsequent more complex evolutions.

Acknowledgement

PT and CJT thank Harvey R. Campbell for his invaluable comments on a previous version of this paper. They also thank Alessandro Recchia and Thayabaran Thanabalasingham for their support and contribution. PT acknowledges the University College London for financial support through the EPSRC program EP/P031730/1. CJT acknowledges the University of Zurich for financial support through the University Research Priority Programme on Social Networks.

Notes and References

[1] Nakamoto, S. "Bitcoin: A Peer-to-Peer Electronic Cash System." (2008) (accessed 29 October 2018) https://bitcoin.org/bitcoin.pdf.

[2] McWaters, R. J. "The Future of Financial Infrastructure: An Ambitious Look at How Blockchain Can Reshape Financial Services." *World Economic Forum* (2016) (accessed 29 October 2018) https://www.weforum.org/reports/the-future-of-financial-infrastructure-an-ambitious-look-at-how-blockchain-can-reshape-financial-services.

[3] Tasca, P. "The Dual Nature of Bitcoin as Payment Network and Money." SUERF Conference Proceedings 2016/1 "Cash on Trial". (2016) http://dx.doi.org/10.2139/ssrn.2805003.

[4] Meguerditchian, V. "Roadmap for Blockchain Standards Report–March 2017." *Standards Australia* (2017) (accessed 29 October 2018) https://www.standards.org.au/getmedia/ad5d74db-8da9-4685-b171-90142ee0a2e1/Roadmap_for_Blockchain_Standards_report.pdf.aspx.

[5] See: http://arstweb.clayton.edu/interlex.

[6] Tasca, P. "Digital currencies: Principles, Trends, Opportunities, and Risks." *SSRN* (2015) (accessed 29 October 2018) https://ssrn.com/abstract=2657598.

[7] Marian, O. "A Conceptual Framework for the Regulation of Cryptocurrencies." *University of Chicago Law Review Online* **82.4** 53 (2015) https://chicagounbound.uchicago.edu/uclrev_online/vol82/iss1/4.

[8] Aste, T., Tasca, P., Di Matteo, T. "Blockchain Technologies: The Foreseeable Impact on Society and Industry." *Computer* **50.9** 18–28 (2017) http://doi.ieeecomputersociety.org/10.1109/MC.2017.3571064.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

[9] Barber, S., Boyen, X., Shi, E., Uzun, E. "Bitter to Better—-How to Make Bitcoin a Better Currency." In A. Keromytis (Ed.), *International Conference on Financial Cryptography and Data Security.* Berlin: Springer 399–414 (2012) https://doi.org/10.1007/978-3-642-32946-3_29.

[10] Walch, A. "The Path of the Blockchain Lexicon (and the Law)." *Review of Banking and Financial Law* **36** 713 (2017) https://www.bu.edu/rbfl/files/2017/09/p729.pdf.

[11] Clack, C. D., Bakshi, V. A., Braine, L. "Smart Contract Templates: Essential Requirements and Design Options." *arXiv* (2016) (accessed 29 October 2018) https://arxiv.org/abs/1612.04496.

[12] Cawrey, D. "Why New Forms of Spam Could Bloat Bitcoin's Block Chain." *CoinDesk* (2014) (accessed 13 January 2017) http://www.coindesk.com/new-forms-spam-bloat-bitcoins-block-chain/.

[13] Xu, X., *et al.* "A Taxonomy of Blockchain-Based Systems for Architecture Design." In *2017 IEEE International Conference on Software Architecture.* 243–252 (2017) https://doi.org/10.1109/ICSA.2017.33.

[14] Otto, K. N., Wood, K. L. "Product Evolution: A Reverse Engineering and Redesign Methodology." *Research in Engineering Design* **10.4** 226–243 (1998) https://doi.org/10.1007/s001639870003.

[15] Bonneau, J., Miller, A., Clark, J., Narayanan, A., Kroll, J. A., Felten, E. W. "SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies." In *2015 IEEE Symposium on Security and Privacy.* IEEE 104–121 (2015) https://doi.org/10.1109/SP.2015.14.

[16] Mattila, J. "The Blockchain Phenomenon–The Disruptive Potential of Distributed Consensus Architectures." *The Research Institute of the Finnish Economy* (2016) (accessed 29 October 2018) https://ideas.repec.org/p/rif/wpaper/38.html.

[17] Wright, A., De Filippi, P. "Decentralized Blockchain Technology and the Rise of Lex Cryptographia." *SSRN* (2015) (accessed 29 October 2018) https://dx.doi.org/10.2139/ssrn.2580664.

[18] Rabbit. "So You Want to Run Rippled?" *Ripple* (2018) (accessed 1 November 2018) https://xrpcommunity.blog/rippled/.

[19] No Author. "History Sharding." *Ripple* (2018) (accessed 1 November 2018) https://developers.ripple.com/history-sharding.html.

[20] No Author. "Validator Details." *Ripple* (2018) (accessed 1 November 2018) https://xrp.media/validator-details/.

[21] Driscoll, K., Hall, B., Sivencrona, H., Zumsteg, P. "Byzantine Fault Tolerance: from Theory to Reality." In S. Anderson, M. Felici, B. Littlewood (Eds.), *International Conference on Computer Safety, Reliability, and Security.* Berlin: Springer 235–248 (2003) https://doi.org/10.1007/978-3-540-39878-3_19.

[22] Castro, M., Liskov, B., *et al.* "Practical Byzantine Fault Tolerance." In *Proceedings of the Third Symposium on Operating Systems Design and Implementation, New Orleans, USA, February 1999.* 173–186 (1999) https://www.usenix.org/legacy/events/osdi99/castro.html.

[23] Cristian, F. "Understanding Fault-Tolerant Distributed Systems." *Communications of the ACM* **34.2** 56–78 (1991) https://doi.org/10.1145/102792.102801.

[24] Fischer, M. J. "The Consensus Problem in Unreliable Distributed Systems (A Brief Survey)." In M. Karpinski (Ed.), *International Conference on Fundamentals of Computation Theory.* Berlin: Springer 127–140 (1983) https://doi.org/10.1007/3-540-12689-9_99.

[25] Tsukerman, M. "Proof of Stake versus Proof of Work." (2015) (accessed 29 October 2018) https://bitfury.com/content/downloads/pos-vs-pow-1.0.2.pdf.

[26] O'Dwyer, K. J., Malone, D. "Bitcoin Mining and its Energy Footprint." In *25th IET Irish Signals and Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014).* (2014) https://dx.doi.org/10.1049/cp.2014.0699.

[27] Lewenberg, Y., Bachrach, Y., Sompolinsky, Y., Zohar, A., Rosenschein, J. S. "Bitcoin Mining Pools: A Cooperative Game Theoretic Analysis." In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems.* Istanbul: International Foundation for Autonomous Agents and Multiagent Systems 919–927 (2015) http://dl.acm.org/citation.cfm?id=2772879.2773270.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

[28] Eyal, I., Sirer, E. G. "Majority Is Not Enough: Bitcoin Mining Is Vulnerable." In N. Christin, R. Safavi-Naini (Eds.), *Financial Cryptography and Data Security 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014.* Berlin: Springer 436–454 (2014) https://doi.org/10.1007/978-3-662-45472-5_28.

[29] Dziembowski, S., Faust, S., Kolmogorov, V., Pietrzak, K. "Proofs of Space." In R. Gennaro, M. Robshaw (Eds.), *Advances in Cryptology–CRYPTO 2015.* Berlin: Springer 585–605 (2015) https://doi.org/10.1007/978-3-662-48000-7_29.

[30] Park, S., Pietrzak, K., Alwen, J., Fuchsbauer, G., Gazi, P. "Spacecoin: A Cryptocurrency Based on Proofs of Space." *IACR Cryptology ePrint Archive* (2015) (accessed 29 October 2018) https://eprint.iacr.org/2015/528.pdf.

[31] Patterson, R. "Alternatives for Proof of Work, Part 2: Proof of Activity, Proof of Burn, Proof of Capacity, and Byzantine Generals." *Bytecoin* (2015) (accessed 25 April 2017, no longer online but available via Internet Archive Wayback Machine) https://web.archive.org/web/20160304055454/https://bytecoin.org/blog/proof-of-activity-proof-of-burn-proof-of-capacity.

[32] No Author. "ripple.com." *Ripple* (2015) (accessed 1 June 2015) http://www.ripple.com.

[33] Barabási, A.-L., Pósfai, M. *Network Science*. Cambridge: Cambridge University Press (2016).

[34] Douceur, J. R. "The Sybil Attack." In P. Druschel, F. Kaashoek, A. Rowstron (Eds.), *Peer-to-Peer Systems.* Berlin: Springer 251–260 (2002) https://doi.org/10.1007/3-540-45748-8_24.

[35] Decker, C., Wattenhofer, R. "Information Propagation in the Bitcoin Network." In *2013 IEEE Thirteenth International Conference onPeer-to-Peer Computing.* IEEE 1–10 (2013) https://doi.org/10.1109/P2P.2013.6688704.

[36] Lamport, L., Shostak, R., Pease, M. "The Byzantine Generals Problem." *ACM Transactions on Programming Languages and Systems* **4.3** 382–401 (1982) https://doi.org/10.1145/357172.357176.

[37] Merkle, R. C. "A Digital Signature Based on a Conventional Encryption Function." In C. Pomerance (Ed.), *Advances in Cryptology–CRYPTO '87.* Berlin: Springer 369–378 (1988) https://doi.org/10.1007/3-540-48184-2_32.

[38] Morrison, D. R. "PATRICIA—Practical Algorithm To Retrieve Information Coded In Alphanumeric." *Journal of the ACM* **15.4** 514–534 (1968) https://doi.org/10.1145/321479.321481.

[39] Wood, G. "Ethereum: A Secure Decentralised Generalised Transaction Ledger." *Ethereum* (2014) (accessed 29 October 2018) https://ethereum.github.io/yellowpaper/paper.pdf.

[40] Buterin, V., Ray, J. "Light Client Protocol: Background: Patricia Merkle Trees." (2014) (accessed 10 November 2018), https://github.com/ethereum/wiki/wiki/Light-client-protocol#background-patricia-merkle-trees.

[41] No Author. "Developer : Block Chain Overview." *bitcoin.org* (2015) (accessed 25 April 2017) https://bitcoin.org/en/developer-guide#block-chain-overview.

[42] Xu, Q., Aung, K. M. M., Zhu, Y., Yong, K. L. "A Blockchain-Based Storage System for Data Analytics in the Internet of Things." In R. Yager, J. Pascual Espada (Eds.), *New Advances in the Internet of Things.* Berlin: Springer 119–138 (2018) https://doi.org/10.1007/978-3-319-58190-3_8.

[43] Tessone, C. J., Tasca, P. "A Parsimonious Model for Blockchain Consensus: Scalability and Consensus Collapse." (2017) Unpublished Working Paper UZH and UCL.

[44] Tasca, P., Widmann, S. "The Challenges Faced by Blockchain Technologies–Part 1." *Journal of Digital Banking* **2.2** 132–147 (2017).

[45] Lombrozo, E., Lau, J., Wuille, P. "Segregated Witness (Consensus Layer)." (2015) Technical Report BIP-141, Bitcoin Improvement Proposal https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki.

[46] Garzik, J. "Segregated Witness in the Context of Scaling Bitcoin." (2015) (accessed 29 October 2018) https://www.mail-archive.com/bitcoin-dev@lists.linuxfoundation.org/msg03051.html.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

[47] Decker, C., Wattenhofer, R. "A Fast and Scalable Payment Network with Bitcoin Duplex Micropayment Channels." In A. Pelc, A. Schwarzmann (Eds.), *Stabilization, Safety, and Security of Distributed Systems, 17th International Symposium, SSS 2015, Edmonton, AB, Canada, August 18-21, 2015, Proceedings.* Berlin: Springer 3–18 (2015) https://doi.org/10.1007/978-3-319-21741-3_1.

[48] Poon, J., Dryja, T. "The Bitcoin Lightning Network: Scalable Off-chain Instant Payments." See: https://lightning.network/lightning-network-paper.pdf.

[49] Lerner, S. D. "Segwit2Mb–combined soft/hard fork–Request For Comments." (2017) (accessed 29 October 2018) https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2017-March/013921.html.

[50] Eyal, I. "The Miner's Dilemma." In *2015 IEEE Symposium on Security and Privacy.* IEEE 89–103 (2015) https://doi.org/10.1109/SP.2015.13.

[51] Tessone, C. J., Garcia, D. "Bitcoin: The Centralisation of a Decentralised Economy." (2017) Unpublished Working Paper UZH.

[52] Catalini, C., Gans, J. S. "Some Simple Economics of the Blockchain." *SSRN* (2017) (accessed 29 October 2018) https://ssrn.com/abstract=2874598.

[53] Sompolinsky, Y., Zohar, A. "Bitcoin's Underlying Incentives." *Communications of the ACM* **61.3** 46–53 (2018) https://doi.org/10.1145/3155112.3168362.

[54] Garcia, D., Tessone, C. J., Mavrodiev, P., Perony, N. "The Digital Traces of Bubbles: Feedback Cycles Between Socio-Economic Signals in the Bitcoin Economy." *Journal of The Royal Society Interface* **11.99** 20140623 (2014) URL https://dx.doi.org/10.1098/rsif.2014.0623.

[55] Tsukerman, M. "The Block Is Hot: A Survey of the State of Bitcoin Regulation and Suggestions for the Future." *Berkeley Technology Law Journal* **30** 1127 (2015) https://doi.org/10.15779/Z38CK4X.

[56] Grinberg, R. "Bitcoin: An Innovative Alternative Digital Currency." *Hastings Science & Technology Law Journal* **4** 159 (2012) https://ssrn.com/abstract=1817857.

[57] Selgin, G. "Synthetic Commodity Money." *Journal of Financial Stability* **17** 92–99 (2015) https://econpapers.repec.org/article/eeefinsta/v_3a17_3ay_3a2015_3ai_3ac_3ap_3a92-99.htm.

[58] Luther, W. J. "Is Bitcoin Intrinsically Worthless?" *Journal of Private Enterprise* **33.1** 31–45 (2018) http://journal.apee.org/index.php/2018_Journal_of_Private_Enterprise_Vol_33_No_1_Spring_parte3.

[59] Luther, W. J. "Cryptocurrencies, Network Eeffects, and Switching Costs." *Contemporary Economic Policy* **34.3** 553–571 (2016) https://doi.org/10.1111/coep.12151.

[60] Buterin, V., *et al.* "A Next-Generation Smart Contract and Decentralized Application Platform." (2014) (accessed 29 October 2018) https://github.com/ethereum/wiki/wiki/White-Paper.

[61] Rosenfeld, M. "Overview of Colored Coins." (2012) (accessed 29 October 2018) https://bitcoil.co.il/BitcoinX.pdf.

[62] Rohr, J., Wright, A. "Blockchain-Based Token Sales, Initial Coin Offerings, and the Democratization of Public Capital Markets." *SSRN* (2017) (accessed 29 October 2018) https://ssrn.com/abstract=3048104.

[63] Adhami, S., Giudici, G., Martinazzi, S. "Why Do Businesses Go Crypto? An Empirical Analysis of Initial Coin Offerings." *Journal of Economics and Business* (in press) https://doi.org/10.1016/j.jeconbus.2018.04.001.

[64] Conley, J. P. "Blockchain and the Economics of Crypto-Tokens and Initial Coin Offerings." *Vanderbilt University Department of Economics Working Papers* (2017) (accessed 29 October 2018) https://ideas.repec.org/p/van/wpaper/vuecon-sub-17-00007.html.

[65] Dilley, J., Poelstra, A., Wilkins, J., Piekarska, M., Gorlick, B., Friedenbach, M. "Strong Federations: An Interoperable Blockchain Solution to Centralized Third Party Risks." *arXiv* (2016) (accessed 29 October 2018) https://arxiv.org/abs/1612.05491.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

[66] Chen, Z.-D., Zhuo, Y., Duan, Z.-B., Kai, H. "Inter-Blockchain Communication." *DEStech Transactions on Computer Science and Engineering* http://dx.doi.org/10.12783/dtcse/cst2017/12539.

[67] Atzori, M. "Blockchain Technology and Decentralized Governance: Is the State Still Necessary?" *Journal of Governance and Regulation* **6.1** https://dx.doi.org/10.22495/jgr_v6_i1_p5.

[68] Davidson, S., De Filippi, P., Potts, J. "Disrupting Governance: The New Institutional Economics of Distributed Ledger Technology." *SSRN* (2016) (accessed 29 October 2018) https://ssrn.com/abstract=2811995.

[69] Wright, A., De Filippi, P. "Decentralized Blockchain Technology and the Rise of Lex Cryptographia." *SSRN* (2015) (accessed 29 October 2018) https://ssrn.com/abstract=2580664.

[70] Tasca, P. "The Hope and Betrayal of Blockchain." (2018) (accessed 9 November 2018) https://www.nytimes.com/2018/12/04/opinion/blockchain-bitcoin-technology-revolution.html.

[71] Kim, H., Laskowski, M. "A Perspective on Blockchain Smart Contracts: Reducing Uncertainty and Complexity in Value Exchange." *arXiv* (2018) (accessed 29 October 2018) https://ssrn.com/abstract=2975770.

[72] Atzei, N., Bartoletti, M., Cimoli, T. "A Survey of Attacks on Ethereum Smart Contracts (SoK)." In M. Maffei, M. Ryan (Eds.), *Principles of Security and Trust, 6th International Conference, POST 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings.* Berlin: Springer 164–186 (2017) https://doi.org/10.1007/978-3-662-54455-6_8.

[73] Details available at https://www.iso.org/committee/6266604.html.

[74] Lin, I.-C., Liao, T.-C. "A Survey of Blockchain Security Issues and Challenges." *International Journal of Network Security* **19.5** 653–659 (2017) https://dx.doi.org/10.6633/IJNS.201709.19(5).01.

[75] Conti, M., Lal, C., Ruj, S., Sandeep, K. E. "A Survey on Security and Privacy Issues of Bitcoin." *arXiv* (2017) (accessed 29 October 2018) https://arxiv.org/abs/1706.00916.

[76] Crosby, M., Pattanayak, P., Verma, S., Kalyanaraman, V. "Blockchain Technology: Beyond Bitcoin." *Applied Innovation Institute* (2016) (accessed 29 October 2018) https://www.appliedinnovationinstitute.org/blockchain-technology-beyond-bitcoin/.

[77] Harvey, C. R. "Cryptofinance." *SSRN* (2016) (accessed 29 October 2018) https://ssrn.com/abstract=2438299.

[78] Meiklejohn, S., Orlandi, C. "Privacy-Enhancing Overlays in Bitcoin." In M. Brenner, N. Christin, B. Johnson, K. Rohloff (Eds.), *Financial Cryptography and Data Security, FC 2015 International Workshops, BITCOIN, WAHC, and Wearable, San Juan, Puerto Rico, January 30, 2015, Revised Selected Papers.* Berlin: Springer 127–141 (2015) https://doi.org/10.1007/978-3-662-48051-9_10.

[79] Ben-Sasson, E., Chiesa, A., Tromer, E., Virza, M. "Scalable Zero Knowledge Via Cycles of Elliptic Curves." In J. A. Garay, R. Gennaro (Eds.), *Advances in Cryptology–CRYPTO 2014.* Berlin: Springer 276–294 (2014) https://doi.org/10.1007/978-3-662-44381-1_16.

[80] Tasca, P., Liu, S., Hayes, A. "The Evolution of the Bitcoin Economy: Extracting and Analyzing the Network of Payment Relationships." *The Journal of Risk Finance* **19.2** 94–126 (2016) https://doi.org/10.1108/JRF-03-2017-0059.

[81] Ben-Sasson, E., *et al.* "Zerocash: Decentralized Anonymous Payments from Bitcoin." In *2014 IEEE Symposium on Security and Privacy.* IEEE 459–474 (2014) https://doi.org/10.1109/SP.2014.36.

[82] de Wit, J. "DAO, Can It Be Viable? An Exploratory Research on the Viability of a Blockchain Based Decentralized Autonomous Organization." *Radboud University* (2017) (Master's Thesis) https://theses.ubn.ru.nl/handle/123456789/4517.

[83] Hearn, M. "Corda: A distributed ledger." (2016) (accessed 29 November 2018) https://www.r3.com/wp-content/uploads/2017/06/corda_technical_R3.pdf.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

[84] No Author. "Mixing Service." *Bitcoin Wiki* (accessed 1 January 2018) https://en.bitcoin.it/wiki/Mixing_service.

[85] No Author. "CoinJoin." *Bitcoin Wiki* (accessed 1 June 2017) https://en.bitcoin.it/wiki/CoinJoin.

[86] Noether, S., Mackenzie, A., Monero Research Lab. "Ring Confidential Transactions." *Ledger* **1** 1–18 (2016) https://doi.org/10.5195/ledger.2016.34.

[87] Möser, M., Böhme, R. "Anonymous Alone? Measuring Bitcoin's Second-Generation Anonymization Techniques." In *2017 IEEE European Symposium on Security and Privacy Workshops.* IEEE 32–41 (2017) https://doi.org/10.1109/EuroSPW.2017.48.

[88] Rivest, R., Shamir, A., Tauman, Y. "How to Leak a Secret." In C. Boyd (Ed.), *Advances in Cryptology—ASIACRYPT 2001.* Berlin: Springer (2001) https://doi.org/10.1007/3-540-45682-1_32.

[89] Guegan, D. "Public Blockchain Versus Private Blockchain." *Université Panthéon-Sorbonne (Paris 1), Centre d'Economie de la Sorbonne working papers* (2017) (accessed 29 October 2018) https://econpapers.repec.org/RePEc:mse:cesdoc:17020.

[90] Maurer, F. K. "A Survey on Approaches to Anonymity in Bitcoin and Other Cryptocurrencies." In H. C. Mayr, M. Pinzger (Eds.), *46. Jahrestagung der Gesellschaft für Informatik, Informatik 2016, 26-30 September 2016, Klagenfurt, Österreich.* Bonn: Gesellschaft für Informatik 2145–2150 (2016) https://dl.gi.de/20.500.12116/1110.

[91] Reid, F., Harrigan, M. "An Analysis of Anonymity in the Bitcoin System." In A. Y., Y. Elovici, A. Cremers, N. Aharony, A. Pentland (Eds.), *Security and Privacy in Social Networks.* New York: Springer 197–223 (2013) https://doi.org/10.1007/978-1-4614-4139-7_10.

[92] Timmerman, K., Thomas, M., *et al.* "Ethereum: More than 'The New Bitcoin'." *The Proctor* **37.5** 26 (2017).

[93] Möser, M., Böhme, R. "Trends, Tips, Tolls: A Longitudinal Study of Bitcoin Transaction Fees." In M. Brenner, N. Christin, B. Johnson, K. Rohloff (Eds.), *Financial Cryptography and Data Security, FC 2015 International Workshops, BITCOIN, WAHC, and Wearable, San Juan, Puerto Rico, January 30, 2015, Revised Selected Papers.* Berlin: Springer 19–33 (2015) https://doi.org/10.1007/978-3-662-48051-9_2.

[94] Carlsten, M. *The Impact of Transaction Fees on Bitcoin Mining Strategies*. (PhD Thesis) Princeton University (2016).

[95] No Author. "Developer Guide: Mining." *bitcoin.org* (2015) (accessed 25 April 2017) https://bitcoin.org/en/developer-guide#mining.

[96] Alvseike, R., Iversen, G. A. G. "Blockchain and the Future of Money and Finance: A Qualitative Exploratory Study of Blockchain Technology and Implications for the Monetary and Financial System." *Norwegian School of Economics* (2017) (Master's Thesis) http://hdl.handle.net/11250/2453330.

[97] Scott, B. "How Can Cryptocurrency and Blockchain Technology play a Role in Building Social and Solidarity Finance?" *United Nations Research Institute for Social Development, Working Paper* (2016) (accessed 29 October 2018) http://www.unrisd.org/brett-scott.

[98] Evans, C. "Bitcoin in Islamic Banking and Finance." *Journal of Islamic Banking and Finance* **3.1** 1–11 (2015) http://dx.doi.org/10.15640/jibf.v3n1a1.

[99] Quintana Diaz, J. M. "The Merger of Cryptography and Finance-Do Cryptographic Economic Systems Lead to the Future of Money and Payments?" *SSRN* (2014) (accessed 29 October 2018) https://ssrn.com/abstract=2536876.

[100] Boudguiga, A., *et al.* "Towards Better Availability and Accountability for IoT Updates by Means of a Blockchain." In *IEEE Security & Privacy on the Blockchain (IEEE S&B 2017) an IEEE EuroS&P 2017 and Eurocrypt 2017 Affiliated Workshop.* (2017) https://doi.org/10.1109/EuroSPW.2017.50.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

[101] Dorri, A., Kanhere, S. S., Jurdak, R. "Towards an Optimized BlockChain for IoT." In *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation.* New York: Association for Computing Machinery 173–178 (2017) https://doi.org/10.1145/3054977.3055003.

[102] Andrews, C., Broby, D., Paul, G., Whitfield, I. "Utilising Financial Blockchain Technologies in Advanced Manufacturing." *University of Strathclyde* (2017) (accessed 29 October 2018) https://strathprints.strath.ac.uk/id/eprint/61982.

[103] Mainelli, M., von Gunten, C. "Chain Of A Lifetime: How Blockchain Technology Might Transform Personal Insurance." *Long Finance and Z/Yen Group* (2014) (accessed 29 October 2018) http://archive.longfinance.net/images/Chain_Of_A_Lifetime_December2014.pdf.

[104] Stellnberger, M. "Insurance Through Blockchain: A Hybrid Approach." *martinstellnberger.co* (2016) (accessed 29 October 2018) https://www.martinstellnberger.co/s/Insurance-through-Blockchain-A-hybrid-approach-2016-v3.pdf.

[105] Badzar, A. "Blockchain for Securing Sustainable Transport Contracts and Supply Chain Transparency: An Explorative Study of Blockchain Technology in Logistics." *Lund University* (2016) (Master's Thesis) http://lup.lub.lu.se/student-papers/record/8880383.

[106] Hackius, N., Petersen, M. "Blockchain in Logistics and Supply Chain: Trick or Treat?" In W. Kersten, T. Blecker, C. M. Ringle (Eds.), *Proceedings of the Hamburg International Conference of Logistics (HICL).* Hamburg: epubli 3–18 (2017) https://doi.org/10.15480/882.1444.

[107] Kotobi, K., Bilén, S. G. "Blockchain-Enabled Spectrum Access in Cognitive Radio Networks." In *2017 Wireless Telecommunications Symposium.* IEEE 1–6 (2017) https://doi.org/10.1109/WTS.2017.7943523.

[108] Plant, L., *et al.* "Implications of Open Source Blockchain for Increasing Efficiency and Transparency of the Digital Content Supply Chain in the Australian Telecommunications and Media Industry." *Australian Journal of Telecommunications and the Digital Economy* **5.3** 15 (2017) https://doi.org/10.18080/ajtde.v5n3.113.

[109] Chakravorty, A., Rong, C. "Ushare: User Controlled Social Media Based on Blockchain." In *Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication.* New York: Association for Computing Machinery 99 (2017) http://dx.doi.org/10.1145/3022227.3022325.

[110] Kuo, T.-T., Kim, H.-E., Ohno-Machado, L. "Blockchain Distributed Ledger Technologies for Biomedical and Health Care Applications." *Journal of the American Medical Informatics Association* **24.6** 1211–1220 (2017) https://dx.doi.org/10.1093/jamia/ocx068.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140

Appendix N: Blockchains Analysed for the Taxonomy

Technology	Description
Bitcoin	Forerunner and by far the most widely used cryptocurrency.
Dash	Privacy-centric digital currency. The digital currency Dash has different functionalities *e.g.*, instant transactions. Dash is based on the Bitcoin source code, but it allows anonymity while performing transactions.
Monero	Monero is an open-source digital currency, with the focus on decentralisation, scalability and privacy. It is based on the CryptoNote protocol, which has an enabled anonymous layer.
LiteCoin	LiteCoin is a P2P cryptocurrency and open source software project. The Litecoin is technically nearly identical to Bitcoin, except for the proof-of-work cryptographic function used.
Zcash	Zcash is a decentralised and open-source cryptocurrency, which combines privacy with selective transparency of transactions.
Peercoin	Cost-effective and sustainable cryptocurrency based on proof-of-stake.
ColorCoin	A concept that allows attaching metadata to Bitcoin transactions and leveraging the Bitcoin infrastructure for issuing and trading immutable digital assets that can represent real world value.
Omnilayer (MasterCoin)	A meta-protocol layer that enables new digital currencies, digital assets, and communication protocol to existing on top of the Bitcoin blockchain.
NameCoin	NameCoin is an asset registry on top of the Bitcoin Blockchain, which enables a decentralised domain name system.
Counterparty	Counterparty enables anyone to write specific digital agreements or programs known as smart contracts, and execute them on the Bitcoin blockchain.
NXT (Ardor)	NXT is a safe, transparent and decentralised system for sharing data and allowing payments to people all over the world. Ardor platforms enable smart contract functions with NXT.
Ethereum	Ethereum is an open-source platform to build blockchain-based applications in different business fields.
Monax	Monax is an open platform for developers and DevOps to build, ship, and run blockchain-based applications for business ecosystems. Monax is known as a private blockchain offered by the monax company.
Cosmos	Cosmos is an architecture for cross-chain interoperability where independent blockchains can interact via an inter-blockchain communication (IBC) protocol, a kind of virtual UDP or TCP for blockchains each driven by the Byzantine fault tolerant (BFT) consensus algorithm, similar to Tendermint.
COMIT	COMIT is a cryptographically-secure off-chain multi-asset instant transaction network (COMIT) that can connect and exchange any asset on any blockchain to any other blockchain using a COMIT cross-chain routing protocol (CRP).
Synerio	Synerio is intended to become a decentralised social network.
Ripple	Ripple is a global real-time financial settlement provider.
Stellar	Stellar is a decentralised multicurrency-exchange platform for people without access to the banking system.
Hyperledger	Hyperledger (or the Hyperledger project) is an open source blockchain platform, started in December 2015 by the Linux Foundation, to support blockchain-based distributed ledgers. It is focused on ledgers designed to support global business transactions, including major technological, financial, and supply chain companies, with the goal of improving many aspects of performance and reliability.
Tendermint	Tendermint is a high-performance blockchain consensus engine that enables to run Byzantine fault tolerant applications written in any programming language. Tendermint is a partially synchronous BFT consensus protocol derived from the DLS consensus algorithm.
Corda	Open-source distributed ledger platform.
2-3 Enigma	Distributed ledger technology, created by the MIT digital currency lab.

Table 1. Blockchain Technologies

SSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.140