IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 28, 2019, accepted January 20, 2020, date of publication January 27, 2020, date of current version February 4, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2969820

Blockchain-Based Distributed Trust and Reputation Management Systems: A Survey

EMANUELE BELLINI[1,2], (Member, IEEE), YOUSSEF IRAQI[1], (Senior Member, IEEE), AND ERNESTO DAMIANI[1], (Senior Member, IEEE)

[1]Center of Cyber-Physical Systems, College of Engineering, Khalifa University, Abu-Dhabi 127788, UAE
[2]Mathematics and Physics Department, University of Campania "Vanvitelli", 81100 Caserta, Italy

Corresponding author: Youssef Iraqi (youssef.iraqi@ku.ac.ae)

This work was supported in part by the Center of Cyber-Physical Systems, Khalifa University, under Grant 8474000137-RC1-C2PS-T3. The work of Emanuele Bellini was supported by the project "Attrazione e MobilitÀ dei Ricercatori" Italian PON Programme (PON_AIM 2018) under Grant AIM1878214-2.

ABSTRACT Distributed Ledger Technologies (DLTs), like Blockchain, are characterized by features such as transparency, traceability, and security by design. These features make the adoption of Blockchain attractive to enhance information security, privacy, and trustworthiness in very different contexts. This paper provides a comprehensive survey and aims at analyzing and assessing the use of Blockchain in the context of Distributed Trust and Reputation Management Systems (DTRMS). The analysis includes academic research as well as initiatives undertaken in the business domain. The paper defines two taxonomies for both Blockchain and DTRMS and applies a Formal Concept Analysis. Such an approach allowed us to identify the most recurrent and stable features in the current scientific landscape and several important implications among the two taxonomies. The results of the analysis have revealed significant trends and emerging practices in the current implementations that have been distilled into recommendations to guide Blockchain's adoption in DTRMS systems.

INDEX TERMS Blockchain, distributed ledger technology, distributed reputation management system, distributed trust management system, formal concept analysis, security, taxonomy.

I. INTRODUCTION

Although manifestations of trust are easy to recognize because we experience and rely on them every day, trust is a multifaceted concept, and its definition can be challenging since it embraces constructs of ethics, morals, emotions, values, and combines a variety of fields.

Moreover, trust is always contextual. For instance, an e-commerce seller can be trusted to sell a product, but cannot be trusted to perform a medical diagnosis. According to Luhmann in [1], trust is an effective mechanism of reducing complexity and risks. The author considers trust as a continuous feedback loop with signals that indicate whether or not the trust is justified. Gambetta in [2], defines trust as the subjective probability by which an individual *A*, expects that another individual, *B*, performs a given action on which its welfare depends.

In [3], it has been noted that trust plays a critical role when a user assesses the believability of online information content or when selecting an exchange site to purchase a product. Users will not believe or participate in a transaction with those whom they do not trust. Thus, trust is defined as the perception of the degree to which an exchange partner will fulfill their transactional obligations in situations characterized by risk or uncertainty. They identify seven dimensions of trust in digital settings: attraction, dynamism, expertness, faith, intentions, localness, and reliability.

In general, trust requires the eagerness of an actor, named the truster, to enter into a position of complexity and uncertainty [1] and thus, become vulnerable inside the relationship with another actor, the trustee. Hence, there are two related prerequisites for trust to arise: risk and interdependence. Indeed, without both conditions, there is no need for trust [4].

Therefore, trust can be seen as a rational form of cooperation under behavioral risk, weighing up possible benefits and costs, and concerning the assessment and management of the risks perceived by each actor entering into a relationship. In other words, "Trust entails Risk" [5]. In Zanini [6], it is explained that, where there is a dearth of reliable evidence, trust is supposed to assure an agent that desirable course of events will be realized in the future as if being guaranteed from experience.

The associate editor coordinating the review of this manuscript and approving it for publication was Leonardo Mostarda.

Reputation is a global perception of an entity's behavior based on the trust that other entities have established [7]. The goal of a trust and reputation management system is then to guarantee that actions taken by entities in a system reflect their reputation values, and to prevent these values from being manipulated by unauthorized entities [8]–[10].

A practical solution is to establish a central authority to intermediate as the agent of trust. In a centralized Trust Management System (TMS)/Reputation Management System (RMS), all the ratings are collected and processed by a centralized controlled computation facility (e.g., cloud). This approach is efficient when the business case requires a trusted third party (e.g., eBay, Airbnb). While the use of a central institution can mitigate the trust problem, it does not eradicate the root cause of mistrust. Moreover, the presence of an intermediary creates new issues of inefficiency, bottlenecks, information asymmetry, and, most of all, the need for the entire system to trust the central authority. The possibility that the central institution might make errors, or even engage in fraud and forgery, requires the centralized institution to make ongoing reconciliation with other centralized institutions (e.g., as was the case during the crisis of trust in the financial sector in 2009). Still, the issues posed by Juvenal in his Satires "Quis custodiet ipsos custodes?" (Who watches the watchers?) remain.

Distributed Trust Management Systems (DTMS) and Distributed Reputation Management Systems (DRMS) have been around almost as long as Peer-to-Peer (P2P) networks themselves, with the first system mentioned in literature in 2003 [11]. Reputation systems in P2P networks all have different goals, including choosing reliable resources, ensuring peers behave honestly, and rating the quality of the content of a shared file. Reputation systems in P2P networks have to contend with the known issues of reputation systems in general, with the additional complexity a P2P network adds. Additional issues such as how to keep reputation data up to date, accurate, and distributed to a broad set of peers which changes dynamically, are faced when deploying such a reputation system.

If the adaption of a distributed approach solved most of the centralized-based approach issues, it is not immune to attacks and potential manipulation. Thus, it is crucial to reinforce the DTMS/DRMS with solutions able to prevent or mitigate the short as well as the long term impact of these attacks. In this regard, the Distributed Ledger Technology (DLT) is emerging as a promising solution. The most famous implementation of DLT is Blockchain (BC).

With the growth of technology, several surveys have been published on DLT to identify its most critical characteristics and building blocks [12]–[17]. The surveys also addressed the applications of DLT that span across different fields, including customer loyalty, cybersecurity, digital rights management, digital voting and government, Internet of Things (IoT), gaming, content distribution, platform development, prediction markets, and Smart Contracts (SCs) [18]–[21].

For instance, in [18], an extensive description of the basics of BC and SCs is provided, and it gives a good overview of the application and deployment of BC-based IoT (BIoT) solutions. The authors in [19] present a holistic approach to BC for IoT scenarios, including not only the basics on BC-based IoT applications, but also a thorough analysis of the most relevant aspects involved in their development, deployment, and optimization.

In [13], a survey on DLT is conducted, however, the focus was limited to its technical implementations. Similarly, in [15], an assessment of the current BC platform against several meta-characteristics of the mode of operation is provided, however, the objective in [15] was the analysis of BC. The authors in [14] provide a generic review of the architecture and the different mechanisms involved in BC without focusing on DTMS or DRMS.

Similarly, several surveys on DTMS and DRMS have been published, such as [22]–[28]. For example, in [22], trust and reputation systems for online service provisioning have been reviewed, while in [28], the focus of the survey is on reputation and trust in multi-agent systems. In [23], the survey was focused on various functional mechanisms and methods to manage reputation and recommendations.

A. MOTIVATION AND KEY CONTRIBUTIONS

As can be noted from the discussion above, even if the TMSs and RMSs seem to be well investigated, at the time of writing, a survey focused on the adoption of BC in TMS/RMS implementation is currently missing in the literary landscape. Therefore, the main contributions of this work are:

1) To survey recent, academic and business, BC-based TMS/RMS systems and assess and analyze the use of BC.
2) To define uniform taxonomies for both BC and TMS/RMS.
3) To identify essential implications among the features of the two taxonomies through Formal Concept Analysis (FCA).
4) To highlight significant trends and provide valuable recommendations for BC adoption in TMS/RMS.

B. PAPER ORGANIZATION

The remaining of this paper is as follows. Section II presents background information relevant to the survey. Section III describes the review methodology adopted in this survey, and Section IV presents the proposed taxonomies for both DTMS/DRMS and BC. Section V presents the surveyed academic papers and business-based applications, and Sections VI and VII present the literature assessment and analysis. Sections VIII and IX present a discussion of the findings and the conclusion respectively.

II. BACKGROUND INFORMATION

A. DISTRIBUTED TRUST AND REPUTATION MANAGEMENT SYSTEMS

There are several examples of DTMS/DRMS. For instance, in [29], a pervasive trust management model, based on trust

relations between entities, is proposed. It suggests that trust can be gained on both direct and indirect basis, and the overall trust score is calculated on a reference model as the average of all recommendations, weighted by the trust degree of the recommender.

Another example is presented in [30], where an integrated trust and reputation model for open multi-agent systems is designed to compute the reputation. In that design, the reputation can be computed based on different sources of trust information: direct experience, witness information, role-based rules, and third-party references provided by the target agents. A bio-inspired trust model for P2P networks based on ant colonies behavior is presented in [31]. In this case, a P2P node requesting a service asks the community and receives a path that leads to the optimal node that offers the requested service.

In [32] and [33], the authors propose to distribute the trust using a distributed hash table structure [34], [35]: every peer holds some part of the information, which allows computing the reputation of a service provider. They also use witnesses for each transaction, which guarantees that the reputation submission will be correctly performed even if one of the two parties was to abort the protocol.

Other protocols such as [36]–[39], and [40] are truly decentralized, and the feedback is retrieved from the participants each time a participant wishes to know the reputation of another participant. Therefore, all the nodes should stay online to contribute to the reputation calculation. This approach has been used in P2P applications but seems to not be suitable for many other domains, such as in e-commerce or IoT. Furthermore, these protocols are rather confidentiality-preserving than privacy-preserving, in that they do not hide the list of users who participated in the rating. This way of partially hiding information leads to multiple issues linked to the mutability of the set of participating peers. For example, the contribution of a user to the aggregated reputation might be revealed if the user goes offline between two reputation-queries. These distributed approaches, however, are not immune to attacks and potential manipulation.

B. LIMITATIONS OF CURRENT DISTRIBUTED APPROACHES

DTMS/DRMS are composed of entities, observers, disseminators, and reputation servers. The authors in [41] have reported that different kinds of attacks can threaten such systems:

- ***Bad-mouthing attack*** consists of lying about the performance of a service provider in order to decrease his reputation, and it is considered the most straightforward attack [42]. In [43], the authors describe a possible defense against such an attack by comparing ratings of users to the ratings of higher trusted users in the network. However, they failed to consider the first attack whereby the peer is selectively malicious [44]. A possible approach is the usage of tokens wherein a customer can only submit a review about the performance of a service provider if he was engaged in a transaction with this service provider. However, this does not entirely mitigate the threat. For instance, the customer could receive an excellent service from the service provider and then lie about the transaction. In this case, outlier detection could be applied [32], although the customer's un-linkability decreases the performance of this method. Hence, in [33], customer linkability has been enforced.
- ***Bad-Collusion attack*** is another famous attack that is common in reputation systems. This attack is based on a group of nodes who collude between each other to lower a target node's reputation or to improve their reputation [45]. One solution against an ongoing colluding attack is to calculate the reputation score based on the average of all reputations received from a peer.
- ***Sybil attack*** consists of a single user that tries to gain access to multiple legal identities for a disproportionately large influence in a reputation system, while arbitrary altering the values. Its success depends on the cost to obtain such an identity [46]. Hence, the risk of a Sybil attack is reduced when the cost to create new identities increases. In fact, the most effective countermeasure is to link the identity to a real-world identity, as described in [47].
- ***Re-entry/Whitewashing attack.*** In this attack, a service provider can choose to behave maliciously. Once they have a low reputation that impacts their attack, they leave that account and generate a new one repeating the cycle all over again. Surprisingly, this attack is efficient not only because of the low cost of entry to the network but also because the system considers a service provider with zero reputation score as higher than a user with negative ratings, providing the user with an incentive to dispose of this account [32]. One way to mitigate this is to keep this attack costly, which could be done by persistently binding the identity of a service provider to, for example, his tax identification number or website, through a specific mechanism (e.g., using a BC). In this solution, the malicious service provider could still change the domain name or the tax identification number, but this would cost money [48].
- ***Ballot stung*** is all about increasing one's reputation. As the service providers generate the tokens that allow feedback-submission on their own, this attack could only partially be mitigated with the use of coins. This limits the number of tokens that can be generated by the service providers. Therefore, if the service provider makes a ballot-stung attack, he would have fewer tokens left for tokens related to ''real'' transactions, and this would cost him money (as the transactions could not take place). Of course, the service provider could still buy tokens with money (if an exchange platform was to be set up), and use them to perform a ballot-stung attack. However, this would again cost the service provider money [49].

The consequence of such attacks may have an additional non-monetary impact, as it will make the participants more reluctant to take a chance on another trustworthy individual [50]. Nowadays, DLT emerges as a promising solution to prevent or mitigate these attacks.

C. DISTRIBUTED LEDGER TECHNOLOGY

A DLT is a technology that implements a ledger in which data is stored across a network of decentralized nodes. The first building blocks for the DLT were from Christian Cachin [51], who proposed a method to use Byzantine agreement as a primitive for implementing atomic broadcast and guaranteeing a total ordering of all delivered messages. The main idea was to create a tamper-proof ledger and to distribute its control among all participants. Contrarily to the centralized network structure, there are no fixed center nodes in BC-based networks. Therefore, no one can change the data recorded in the ledger unless he has obtained enough capacity to get control of the system. Hence, DLTs implement a level of transparency, traceability, and security by design [52]. These properties make them suitable to be a technology that enhances security, privacy, and trustworthiness in very diverse contexts such as in IoT [53]–[55], voting system [56], machine-to-machine environments [57], and persistent identifier systems [58].

At their core, DLTs are data structures where transactions can be recorded, and a set of functions to manipulate them. Although each DLT differentiates itself using different data models and technologies, all DLTs are based on three well-known pillars, i) public-key cryptography, ii) distributed P2P networks, and iii) consensus mechanisms [13].

The challenge of reaching agreement among multiple members in a network, where there is only limited trust between them, is well described by the so-called ''Byzantine General's Problem'', and was first proposed by computer scientists in 1982 [59]. The difficulty in assessing the trustworthiness of the messages exchanged among the generals led the researchers to conclude that the distributed consensus cannot be reached in an asynchronous fault-tolerant computational model with one faulty process [60].

In order to solve such a problem in the Internet, several consensus techniques have been proposed. The most famous one is the Proof-of-Work (PoW), which is a randomization process based on a competition that solves a mathematical hash puzzle. Because it is merely a process of computational iteration, finding the solution becomes a matter of effort, rather than a matter of mathematical ability [61]. The most widely criticized point of Bitcoin's PoW consensus protocol is the significant amount of energy that is consumed in the mining process. According to Deetman, Bitcoin could consume as much electricity as Denmark by 2020 [62]. Such a resource-intensive nature prevents Bitcoin's style of PoW consensus from being used in other application areas.

Alternative consensus mechanisms, such as Proof-of-Stake (PoS), Paxos-based consensus [63], and Byzantine Fault Tolerance (BFT)-based consensus [64], have been determined to be suitable for improving the efficiency of the second-generation of DLT and cryptocurrency [65]. PoS, in particular, aligns the incentives of digital currency holders in the BC with the good operation of the BC. The aims are to reduce electricity consumption, to improve scalability, and to reduce the so-called 50% +1 attack (if an attacker gets the control of 50% +1 of the nodes in the network, then he will become able to take over the entire system). Validators under a PoS system will have their funds tied up as a stake in the network, which means that it is in their interest to act in the best interest of the network as a whole. There are various PoS protocols such as Tendermint[1] used in Eris and Casper as the amalgamation of two research projects, which are currently being undertaken by the Ethereum development team: Casper the Friendly Finality Gadget and Casper the Friendly GHOST: Correct-by-Construction.

The BFT-based protocol Practical Byzantine Fault Tolerance (PBFT) [66] achieves consensus in an environment where communication is asynchronous, message delays are bounded, and the number of Byzantine servers f is at most $f < n/3$, where n is the number of servers. BFT-based BCs offer a much stronger consistency guarantee and a lower latency, but only for small scale state machine replication scenarios. That is one of the reasons why it is used in permissioned BC as Hyperledger[2] (pluggable) or Stellar.[3] Raft [63] is a protocol that is based on Paxos [67] and implements a deterministic algorithm in asynchronous systems and tolerates $f < n/2$ crash recovery failures, where f is the number of faulty nodes, and n is the total number of nodes in the network. Raft is considered as efficient as Paxos with a structure that is more understandable and closer to real system implementation [68].

Despite these works and others on protocols, studies reveal the existence of several limitations [69], [70]. In [70], a comprehensive survey on Bitcoin technology was presented. The review of existing Proof-of-X consensus protocols pointed out the strengths and weaknesses of each consensus scheme, and concluded that it is still unclear which Proof-of-X approach is most promising to improve Bitcoin and which one will survive in practice.

Even so, a consensus process may still fail. In this case, the content hosted on the BC may be subjected to loss of integrity or loss of consistency [71]. Consensus algorithm failure may take several forms, some of which are:

- Fork: This situation occurs when a set of nodes converge towards a different chain than the rest of the network. Fork sometimes happens naturally in a given BC due to latency. Most of the time, temporary forks are resolved within a period of 2 to 3 block times [71]. Even if a ''fork'' is considered a feature of BC, to insure integrity and consistency, the consensus protocol should resolve any fork rapidly

[1] https://tendermint.com/
[2] https://www.hyperledger.org/
[3] https://www.stellar.org/

under the penalty of creating a final fork or network partitioning.

- Lack of consensus: This is a situation where the nodes are not able to reach an agreement on the current state of the system hence, the transactions. In this case, the BC becomes ineffective.
- Domination: This is a situation where an attacker creates a large number of pseudonymous identities or nodes. The attacker can then gain a disproportional influence to confuse the network and manipulate the consensus towards a given goal.
- Cheating: This is a situation where a node or a set of nodes willingly maintain a parallel chain. This attack is used to present to a participant (e.g., a service provider) a parallel reality that does not exist in the BC.
- Poor performance: Depending on the consensus algorithm, network latency, network instability, malicious nodes, and the complexity of the SC/chaincode, the nodes may require more time to process a transaction and converge towards a single chain.

Moreover, as highlighted in [13], there are several kinds of DLTs. They are currently implemented with the following technologies: BC [72], Tangle [73], Hashgroup, and Sidechain [13]. Among the implementations of DLT, the most famous and mature, is BC. It comes out with Satoshi Nakamoto's proposal [72] to respond to the crisis of trust that occurred in the financial sector during 2007 because of the centralized control system [13]. Today, BC is being adopted in several diverse domains. However, in this paper, we focused only on the solutions utilizing BC technology in DTMS/DRMS.

The main features of BC are:

- Decentralization: In centralized network infrastructures, data exchanges (i.e., the transactions) are validated and authorized by trusted central third-party entities. This incurs costs in terms of centralized server maintenance, as well as performance. In BC-based infrastructures, two nodes can engage in transactions with each other without the need to place trust upon a central entity to maintain records or perform authorization.
- Immutability: Since all new entries made in the BC are agreed upon by peers via decentralized consensus, it is nearly impossible to tamper with the BC, and it is censorship-resistant. In fact, all previously held records in the BC are also immutable, and in order to alter any previous records, an attacker would need to compromise a majority of the nodes involved in the BC network. Hence, any changes in the BC contents are easily detected.
- Auditability: All peers hold a copy of the BC, and thus, can access all timestamped transaction records. This transparency allows peers to look up and verify transactions involving specific BC addresses. Because in real life, BC addresses are not associated with identities, it provides a manner of pseudo-anonymity.

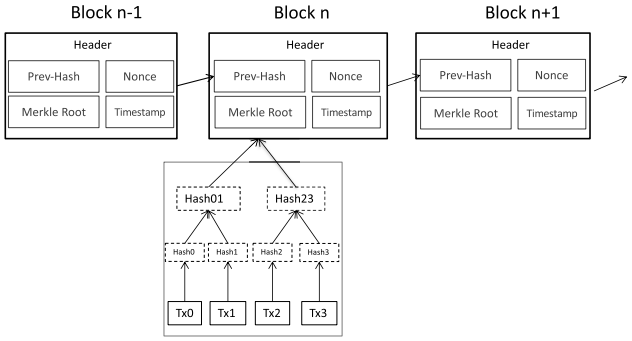

FIGURE 1. Blockchain schema.

Although records of a BC address cannot be traced back to the owner, specific BC addresses can indeed be held accountable and inferences can be made about the transactions made by a specific BC address.

- Fault tolerance: All BC peers contain identical replicas of the ledger records. Any faults or data leakages that occur in the BC network can be identified through decentralized consensus, and later, data leakages can be mitigated using the replicas that are stored in BC peers.

The main building blocks of a BC are:

- Transactions: which are signed pieces of information first created by the participating nodes in the network then broadcasted to the rest of the network. Transactions are encrypted using mathematical algorithms and shall be verified before being hashed and encoded into a Merkle tree whose root is the hash of the considering block [74].
- Blocks: which are collections of transactions that are appended to the BC after being validated. Each block contains a timestamp, a unique ID (i.e., the hash of the Merkle tree), and the ID of the previous block which acts as a link between them (Fig. 1).
- Distributed ledger: which contains all the created blocks that make up the network. Valid transactions will first be appended to the existing chain of blocks, then synchronized, and finally distributed across the network. As a result, every node in the network has the same copy of the database.
- Hashing mechanism/Public keys: which connect the different blocks consecutively. By nature, BC is inherently resistant to data modification. Once recorded, data in any given block cannot be altered retroactively as this would invalidate all hashes in the previous blocks in the BC [75].
- Consensus mechanism: which is used to decide which blocks will be added to the BC. In BC, a transaction is considered valid only if more than 50% of the nodes in the network reach a consensus about its validity following the principle of the *longest chain wins* [72].

D. SMART CONTRACTS

N. Szabo was first to introduce the term smart contract and defined it as a "*computerized transaction protocol that*

executes the terms of a contract'' [76] with the objective of ''securing relationships on public networks'' [77] minimizing the need for trusted intermediaries, and the occurrence of malicious or accidental exceptions [18]. A smart contract is a Turing-complete and deterministic program that includes an arbitrary executable script and a data model that is saved in a BC as a Merkle hash tree. Through exposing public functions, a smart contract interacts with users to offer predefined business logic. Moreover, a smart contract can implement Read-Modify-Write operations altering data in the BC and save the result of the processing in the BC itself. A smart contract can also be executed in a read-only mode (e.g., computational contracts).

A smart contract is triggered by addressing a transaction to it. It has the ACID (Atomicity, Consistency, Isolation, Durability) properties [78] and executes independently and automatically on all or a subset of peer nodes in the BC network. Examples of BCs that use SCs (or chaincode) to interact with the users and the ledger are Ethereum, Hyperledger, Cardano, and EOS. Ethereum is undoubtedly the most well established one. SCs can be programmed with different programming languages. For instance, Ethereum uses Solidity [79], while in other solutions such as Hyperledger, there are more options (e.g., GO, JavaScript, and Node.js).

It is worth to note that even SCs are affected by vulnerabilities. In [80], the stealing of $50 million in funds from the Decentralized Autonomous Organizations - DAO [81] has been analyzed. The attacks exploited a concurrency-based vulnerability (latency in updating the amount of an account after an operation) of a contract since the SC was designed and implemented as a simple single-threaded program. In this respect, in [82], a survey of pitfalls and common bugs in SCs, that are disguised versions of common concurrency pitfalls, is provided. While in [83], several vulnerabilities in Ethereum SC design are also analyzed. These studies raise awareness and call for a general shift in BC from the enthusiastic early adoption towards a more professional and engineered approach.

III. REVIEW METHODOLOGY

Systematic reviews and meta-analysis are a central element of evidence-based analysis. In [84], a review is considered systematic if it is based on clearly formulated questions, identifies relevant studies, appraises their quality, and summarizes the evidence by use of an explicit methodology. The approach is structured in five steps:

- Step 1: Framing questions for review
- Step 2: Identifying relevant work
- Step 3: Assessing the quality of studies
- Step 4: Summarizing the evidence
- Step 5: Interpreting the findings

A. FRAMING QUESTIONS FOR REVIEW

The problems to be addressed by the review should be specified in the form of clear, unambiguous, and structured questions before beginning the review work. To this end, the following questions have been identified:

- What are the intended achievements/improvements behind the use of BC in DTMS/DRMS implementations?
- What are the data stored in the distributed ledger?
- What are the methods to compute trust?
- What is the cost (if any) for feeding a reputation score to the ledger?
- What are the consensus methods used in the BC for the DTMS/DRMS?
- Which machines or nodes in the network are computing the reputation calculations?
- How frequently are the reputation calculations occurring?

Answering these questions allows better identification of the BC technological features fulfilling the DTMS/DRMS functionalities.

B. IDENTIFYING RELEVANT WORK

The study selection criteria should flow directly from the review questions and be specified *a priori*. Because the field of research is very young, but at the same time, the early adopters are continuously increasing, the following kind of materials has been considered:

- Journal/conference papers (peer-reviewed)
- Technical reports published in university/lab series
- White papers from current business implementations (e.g., Initial Coin Offering - ICOs)

C. ASSESSING THE QUALITY OF STUDIES

Study quality assessment is relevant to every step of a review. Question formulation (Step 1) and study selection criteria (Step 2) should describe the minimum acceptable level of design. Selected studies should be subjected to a more refined quality assessment by the use of general critical appraisal guides and design-based quality checklists (Step 3). In this respect, we selected the contribution that presented practical experiments, while merely speculative articles (e.g., position papers) or papers with limited discussion on the use of BC have not been included in the present review. This criterion made it possible to exclude about 30% of the retrieved works.

D. SUMMARIZING THE EVIDENCE

These detailed quality assessments, carried out in Step 3, are used for exploring heterogeneity and forming decisions regarding the suitability of meta-analysis (Step 4). Also, they help in assessing the strength of inferences and making recommendations for future research (Step 5). Data synthesis consists of tabulation of study characteristics, quality, and effects, as well as the use of statistical methods for exploring differences between studies and combining their effects (meta-analysis). In the present work, the analysis of the literature has been performed using the FCA method [85]. In FCA, a concept is composed of an object and its essential features.

Thus, the concepts arise from the data that is given by a formal context. The set of all concepts emerged by a formal context form a mathematical structure named concept lattice. A formal context is defined as:

$$K = (O, P, S)$$

where O is a set of objects, P is the set of properties, and

$$S \subseteq O \times P$$

is the relation that connects each object o with its property p [86]. Given a set of objects $C \subseteq O$, C' is called the intent of C and consists of all the set of attributes that are satisfied by all the objects in C.

$$C' = \{p \in P \mid \forall c \in C; (c, p) \in S\}$$

Thus, the intent comprises all attributes shared by those objects. In the same way, given a set of properties $D \subseteq P$, D' is called the extent of D and consists of all the objects having all the properties in D.

$$D' = \{o \in O \mid \forall d \in D; (o, d) \in S\}$$

Thus, a *formal concept* of a given context is defined by the pair (C, D), where $D' = C$ is called the extent, and $C' = D$ is called the intent. This definition is aligned with the ISO 704, which considers the concept as "[...] a unit of thought constituted of two parts: its extent and its intent.".

The use of FCA to analyze literature is an emerging method that has been proposed in [87], where the literature has been represented as a concept lattice in which the objects are the scientific papers and the attributes are the relevant terms present in the title, keywords, and abstract of the papers.

In the present work, the resulting concept lattice is formed as:

- O: the paper presenting the use of BC to implement a solution of DTMS/DRMS,
- $P = P_{bc} \cup P_{dtms}$ where P_{bc} are the properties defined by the BC taxonomy, and P_{dtms} are the properties defined by the DTMS/DRMS taxonomy.

E. INTERPRETING THE FINDINGS

The interpretation phase consists of assigning a meaning to the results of the literature analysis. The primary intent is to derive the best practices and future trends in order to provide a useful recommendation to the practitioners. In particular, several important logical implications between BC properties and DTMS/DRMS properties, that emerged from the FCA analysis, have been analyzed and explained.

IV. TAXONOMIES

One of the outcomes of this work is to highlight the essential features of BC that are exploited in DTMS/DRMS implementations. Thus, to organize knowledge on BC and DTMS/DRMS, we adopted the taxonomy development approach proposed in [88], where taxonomy is defined as a set of dimensions mutually exclusive and collectively exhaustive in a way that each object under consideration has one and only one. The criteria identification has been inspired by the components and issues identified in [89] for Reputation Systems in P2P networks.

A. DISTRIBUTED TRUST AND REPUTATION MANAGEMENT SYSTEMS TAXONOMY

In [90], the authors classify DTMS into three categories:

- Credential-based trust management systems: in these systems, service providers and the provided services are trusted, but service requesters are not. Service providers use credentials to evaluate the trustworthiness of service requesters, and services may be granted or not.
- Reputation-based trust management systems: in these systems, service providers and provided services are not trusted, and service requesters select service providers based on their reputations.
- Social network-based trust management systems: these systems are based on social networks. Reputation is computed based on social relationships.

The work in [13] is focused on reputation computation features identifying some options such as simple summation, Bayesian systems, discrete trust models, belief models, and fuzzy models. In [91], a general overview of reputation systems applied in P2P networks is provided. A taxonomy for reputation is provided in [92]. The first level of the taxonomy distinguishes between explicit and implicit reputation systems. For the authors, implicit reputation represents systems that have not defined a reputation system, although reputation information exists among its members to assist in making decisions. An example is the common word of mouth system. Explicit reputation is something that has been purposely implemented to facilitate the estimation of trust between members of an environment. This second one is implemented through systems defined by the following 14 dimensions: history, context, collection, representation, aggregation, entities, presence, governance, fabric, interoperability, control, evaluation, data filtering, and data aging. Some of these dimensions have been considered relevant in the present work, such as aggregation, interoperability, control, and data aging, and are included in the proposed taxonomy.

In [23], eleven criteria are provided to cover three topics: 1) the creation and content of a recommendation, 2) the selection and use of recommenders, and 3) the interpretation and reasoning applied to the gathered information. In this context, it is valuable to understand to what extent the characteristics of DTMS/DRMS have been implemented with a BC.

In this section, we propose a taxonomy of the DTMS/DRMS properties (Fig. 2). For each property, we identified mutually exclusive values that represent a DTMS/DRMS feature. For each property and its related features, a description is provided.

A1-Information. This property represents the type of information collected and managed by the DTMS/DRMS.

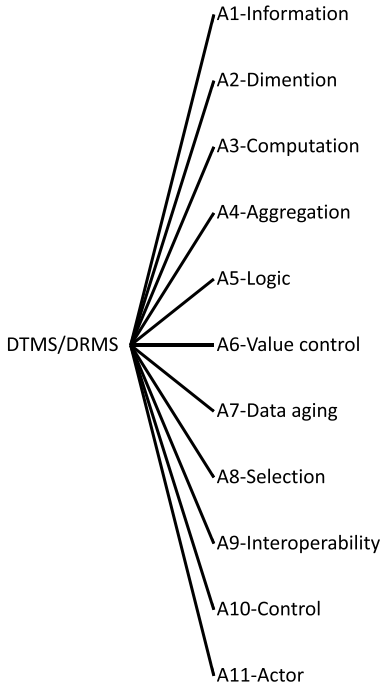

FIGURE 2. DTMS/DRMS Taxonomy.

- *A1.1 Transaction + reputation*. This value is assigned to the solutions that integrate both transactions (assets trade tracking) and the related reputations.
- *A1.2 Transaction score only*. The feature defines a system that manages only the results of the transactions occurred either online or offline.
- *A1.3 X as a transaction (indirect reputation)* (vote, obligation, certificate). This feature identifies solutions adopted in a context where the transaction is not represented by an economic-based good or service trade. In particular, the transaction is represented by a vote expressed, or a certificate earned. The reputation is then calculated or derived based on these assets (e.g., the number of received votes).
- *A1.4 Trust/Reputation score only*. This feature is related to a system that manages the reputation scores only. Eventually, it describes the trust computation results.

A2-Dimension. This property allows discriminating those systems that are domain-oriented (single dimension) versus those that are general purpose and can manage multiple aspects. The concept of trust is always related to a behavior or a context. A high level of reputation and trust obtained in a context or through a behavior should not be transferred to another context. For instance, being a good doctor should not imply being a good manager. The reputation values of a person acting as a doctor and as a manager need to be managed separately.

- *A2.1 Complex/Multiple*. It refers to the possibility of managing the trust of multiple actions or conditions.
- *A2.2 Single*.

A3-Computation. The computation of trust in a BC-based DTMS/DRMS can be performed in different ways. The difference may impact the service performance and the value of the trust at a specific instant.

- *A3.1 Complete history calculation (not lossy)*. It is about the calculation of the complete agent transaction history whenever it has been requested. These calculations would include all variables affecting reputation over a predefined time, and if this period is too long, the calculations become very computationally expensive.
- *A3.2 Transactional update (not lossy)*. This approach implies to keep the reputation value for an agent in the network continuously up to date. This may include having every BC transaction carry reputation data or in case of the SC approach, a data model where the values are stored and paired with the agent of reference.
- *A3.3 Period-based history (lossy)*. Calculate the reputation only once and for a fixed period. An agent's reputation would not change until that period is over. This method is effective and computationally cheaper than complete recalculation. At the start of a new period, the reputation scores would be very accurate. However, the accuracy towards the end of the period would suffer since none of the current behavior would be a factor until the period is complete and reputations are recalculated. This approach can be used in combination with features A7.2 and A7.3.

A4-Aggregation. The work in [92] describes the methods by which a reputation score is computed. Several methods exist in the literature designed to support a specific kind of application properly. In this property, we focus on what we consider the most common approaches.

- *A4.1 Deterministic*. A classical deterministic approach represented by the summation of all of the positive and negative ratings, as presented in [22].
- *A4.2 Probabilistic*. This approach may adopt probabilistic techniques (e.g., Bayesian network, Maximum Likelihood Estimation) to calculate the reputation value based on the probability of peers to provide reliable information [89].
- *A4.3 Flow models*. Reputation is computed by examining the flow of transitive trust. [22]

A5-Logic. This property represents the relevant logic behind the computation of the reputation.

- *A5.1 Bad transaction (behavior) vs. good transaction (behavior)*. The calculation mechanism takes into account both bad and good transactions in the computation.
- *A5.2 Local trust data vs. reputation gathered from other peers*. The reputation and trust values are calculated, taking into account both subjective and peers' perspective.
- *A5.3 Recent transaction vs. old transaction*. This property includes in the computation the timing of when the transaction occurred. In this way, it is possible to weight

the transactions differently depending on the moment they occurred.

- *A5.4 Good transaction or bad transaction count*. This calculation mechanism looks at only good or bad transactions. This means that the reputation/trust value is calculated accumulating a one dimension feedback only. The value grows based on the number of transactions and the stability of the behavior (positive or negative).

A6-Value Control. This property defines the actor that manages trust computation. The value of trust can be collectively accepted or subjective. In the first case, the community should revise and agree on the calculated value, while for the latter, the user is the only one entitled to calculate and modify the trust value that will represent its subjective point of view.

- *A6.1 Community*. This feature belongs to the systems that consider the value of trust assigned to an actor commonly accepted and valid in the entire system.
- *A6.2 Client*. A system that allows a single actor to compute its reputation and trust values for the actors with which it interacts has this feature.

A7-Data aging. This property has been presented in [92]. It is considered a mechanism to calculate the reduction of the confidence of information as time passes. A decay function may assign a heavier weight to recent behavior and a lighter one to transactions far in time. Hence, it is possible to mitigate attacks on the reputation system in which an entity with a high level of trust starts to act maliciously since the level of trust will be rapidly reduced. It is also possible to neglect, from the trust computation, those transactions that are beyond a certain threshold in time.

- *A7.1 None*. No decay function is considered. Thus, we assume that reputation information is retained indefinitely.
- *A7.2 Decay*. A decay function is considered
- *A7.3 Dead of old/selected*. Information is discarded on a time-based threshold or discarded based on specific criteria.

A8-Selection. This property describes the method used in DTMS/DRMS to select a peer or to decide to start a transaction with a counterpart.

- *A8.1 Ranked based*. The transaction is triggered based on the level of reputation/trust of the counterpart compared to the other participants in the network.
- *A8.2 Threshold approach*. The transaction is triggered if the reputation/trust value obtained by the counterpart is above a predefined value.
- *A8.3 Probabilistic based*. The transaction is triggered with a probability proportional to the reputation of the counterpart.

A9-Interoperability. According to [92], the interoperability property is related to the scope of the system. Commercial reputation systems are tightly controlled, and the information contained within them is not shared with third parties, nor can it be reused since they are valid only within the system. This means that good reputations obtained in a system are not portable to another one. On the contrary, interoperable systems allow such portability.

- *A9.1 Open*. Entities may freely access and utilize the reputation and trust information contained within a system, using data standards or application programming interfaces.
- *A9.2 Closed*. Reputation information is proprietary and not usually shared outside the system (e.g., commercial systems).

A10-Control. According to [92], *control* describes how a reputation system motivates and controls entities to act in a desired manner, and is a fundamental aspect of any implementation. This dimension is concerned with explicit rules and incentives/disincentives used within a reputation system in order to get entities to behave in a desired manner.

- *A10.1 Incentives*. An entity is motivated or guided using rewards and punishments to obtain appropriate behaviors.
- *A10.2 Rules*. An entity is forced or limited to act only within a prescribed manner.

A11-Actor. This property discriminates if the actors involved in the system are humans or machines (e.g., sensors, robots, computers).

- *A11.1 Human*. The actors considered in the system are people. They include organizations, service providers, and clients.
- *A11.2 Machine*. Here the actors are devices that interact with each other like vehicles in a vehicular ad hoc network (VANET), robots, and sensors in IoT.

B. BLOCKCHAIN TAXONOMY

Currently, several taxonomies on BC aim to be exhaustive trying to capture and systematize all BC properties, in order to guide BC-based application implementations, as in [12], [13], [16], [93], and [94]. According to [16], isolated knowledge of technical and application research causes hypes of BC application areas and technical BC characteristics. Apart from its high-level properties and generic building blocks presented in Section II-C, BC is a multifaceted technology that can be implemented in different configurations to better address the business case under consideration. Thus, different features may be relevant according to the application. For instance, a study comparing digital payment providers identifies permissions to read and write financial transactions as important technical characteristics to consider when choosing between centralized and decentralized payment platforms [95]. Similarly, a review on cryptocurrencies investigates different properties considered relevant for them, such as consensus mechanisms, levels of anonymity, and data integrity [96]. The analysis of consensus mechanisms like PoS or PBFT is dedicated to verifying their capacity to improve the efficiency of second-generation crypto-currencies [65], [97], [98]. Other works are specialized on token classification [99] that is relevant in all the cases involving tokenization of the transaction.

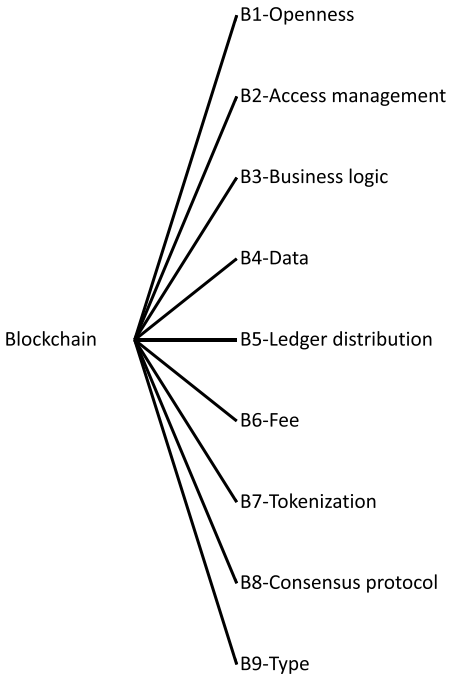

FIGURE 3. BC Taxonomy.

In this respect, we propose a taxonomy of BC properties considered relevant to implement BC-based DTMS/DRMS (Fig. 3).

For each property, we identified mutually exclusive values that represent a BC feature. For each property and its related features, a description is provided.

B1-Openness. This property relates to the governance of the BC implementation.

- *B1.1 Public*. A public BC is a BC that anyone in the world can read, send transactions to, expect to see them included if they are valid, and participate in the consensus process - the process for determining what blocks get added to the chain and what the current state is [100].
- *B1.2 Private*. This property relates to both fully private and consortium-based BC. The former is a BC where write permissions are kept centralized by one organization. Read permissions may be public or restricted to an arbitrary extent. The latter is controlled by a set of pre-selected nodes, which typically belong to multiple organizations. The permission settings also control access to the BC. A consortium BC can be considered as partially decentralized [100].

B2-Access management. This property is related to the read/write rights for the BC participants.

- *B2.1 Permissionless*. Permissionless BC enables anybody to take an interest. The exchanges are approved and handled by votes/agreement. A vote does not rely upon having an earlier character of any sort inside the record, and no previous trust is expected between interested hubs. Examples are Ethereum and Bitcoin.
- *B2.2 Permissioned*. Permissioned BCs limit access regarding who can perform different activities on the BC. Examples are Ripple and Hyperledger. According to [101], permissioned BCs are resource-efficient and easy to maintain and upgrade, as they avoid the need for resources spent on achieving consensus, by limiting the number of untrusted entities that can write in the BC.

B3-Business Logic. This property refers to the possibility of managing the business logic of a DTMS/DRMS within the BC (e.g., using SCs) or outside of it.

- *B3.1 On-chain*. At least one node in the BC network executes reputation software (SCs/chaincode) and would require these nodes to be financially motivated to do so.
- *B3.2 Off-chain*. Applications implementing the business logic are running outside the BC. Since this would rely on an external system, it is faster to deploy, but presents several security risks. Since these nodes are separate from the parent BC, it may also complicate the reward or payment system.

B4-Data. This property is related to how the reputation and trust values are treated in the BC.

- *B4.1 Embedded in the transaction*. In Bitcoin, there exists OP_RETURN, a script opcode used to mark a transaction output as invalid, which is also used to store arbitrary data in the BC. The OP_RETURN lets developers associate up to 80 bytes of arbitrary data with their transactions by adding an immediately prune-able zero-valued output instruction. This instruction returns immediately with an error so that the included data is not interpreted as a script and cannot be used as an input for another transaction.
- *B4.2 Payload transaction*. Custom implementations of BC may decide to modify the structure of the block in order to accommodate a specific type of information.
- *B4.3 Smart contract variables*. Ethereum, as well as Hyperledger and others, provide a mechanism to store arbitrary data using SCs. The data model can be elaborate or linear and can be defined with JSON-DL notation, as in Hyperledger. The cost of managing data with the contract could be diverse, according to the BC business model. In Ethereum, the cost is based on the number of SSTORE operations on the contract variable.

B5-Ledger distribution. This property relates to the level of replication of the ledger in the BC-based DTMS/DRMS.

- *B5.1 Full node*. All the BC actors should deploy the entire full node. The full ledger is distributed to all actors participating in the BC and allowed to interact with the network. It is worth to notice that, as the BC grows, the time required to synchronize a new node increases.
- *B5.2 Miner full + actor thin*. This approach foresees that the actor may interact with the BC without the need to replicate the full ledger.

B6-Fee. The property considers the presence of a fee mechanism to interact (making transactions) with the BC.

- *B6.1 No fee (not defined or optional)*. This feature is related to the fact that the system does not explicitly require the actors to pay for executing transactions.
- *B6.2 Present (variable or fixed)*. The feature refers to the presence of a fee (variable or fixed) to execute a transaction. Variable fees can be calculated based on the number of assets, coins, or data in the SC that is exchanged in the transaction. While the fixed fee is related to the occurrence of a cost that is linked to the transaction itself, and is independent of any other parameters.

B7-Tokenization. This property is related to the reputation and trust representation as a token in the BC. Several standards define the characteristics of tokens, such as:

- Ethereum Request for Comment (ERC)1329: Inalienable Reputation Token #1329. Reputation tokens are emitted and burned by a contract depending on balance holders' actions and their consequences (i.e., there is a proof of the Byzantine behavior of the owner). Reputation balances can be queried, but not directly changed or transferred from outside of the contract. The assumption is that if the reputation may be transferred, it can be sold. However, if it can be sold, it breaks the whole game setting and economic incentives, creating different Nash equilibrium and generally getting worse protection from malicious actors (lower Byzantine tolerance).[4]
- ERC20 specification can improve the interchangeability of ERC20-based tokens and perform the same operation on the Distributed App (DApp). ERC20 avoids the problems of users of the Ethereum community, creating unique tokens and functions, solving the problems of destroying SCs, and hacking attacks during token transfers.
- ERC721 is a popular specification other than ERC20. The most significant difference between ERC721 and ERC20 is that ERC721 defines non-interchangeable tokens, which means that each token has an independent ID, so the independence of ERC721 can be used in the transaction of assets and tracking.

Hence property B7 can be divided into two features.

- *B7.1 No tokens*.
- *B7.2 Present*. This feature includes both transferable and non-transferable tokens.

B8-Consensus protocol. In a BC-based system, users can read or write to the ledger without the control of a trusted third party. The state of the system is the result of an agreement of the nodes, called consensus. In addition to the consensus protocols presented in this section, there are several other consensus mechanisms like Proof of Elapsed Time, Proof of Authority [102], Proof of Burn [103], and Proof of Importance [104]. These are not listed as separate properties because they have not been used in any BC-based DTMS/DRMS surveyed work.

[4]https://github.com/ethereum/EIPs/issues/1329

- *B8.1 Proof-of-Work*. The Bitcoin system broadcasts transactions to all the nodes in the network. Each node collects the new transactions into a block and then works on finding a difficult "Proof-of-Work" for its block, which is called the "mining" process. The mining process involves scanning for a value when it is hashed with SHA-256, the result begins with several zero bits. Since the average scanning work required is exponential in the number of zero bits required but can be verified by simply executing a single hash, it is a perfect approach to presenting PoW. The number of zero bits is used as a parameter to adjust the difficulty, i.e., the average time that a block is found, which normally is around 10 minutes. The finality of a given block increases each time a new block is built (i.e., validated and added to the BC) on top of a previous block. The finality can be expressed as $f(t) = (1 - (A/(1 - A))^{t/b})$. Where $f(t)$ is the probability of finality at time t, A is the ratio of corrupted nodes (in hashing power) $[0 : 1]$, $t > 0$ is the time in seconds, and b is the block time. t is discrete and can only be a multiple of b.
- *B8.2 Proof-of-Stake*. The proof of stake was created as an alternative to PoW, to tackle underlying issues like large energy consumption, or centralization resulting from industrial scaling. PoS mechanism allocates the probability for a given participant to be elected as the leader to update the ledger according to how many tokens the participants hold. For example, someone holding 1% of the tokens has a probability of 1% to validate a block. Early versions of PoS highlighted that "there is nothing at stake" [105]. Indeed, nothing prevents a node from voting for multiple chains since this misconduct would be the optimal strategy from a self-economic point of view. New PoS consensus introduces the notion of slashing. To participate in the consensus, each node must provide a security deposit. In the event of misconduct, e.g., double chain validation, the safety deposit would be slashed [105]. The finality f is probabilistic. The finality of a given block increases each time a new block is built (i.e., validated and added to the BC) on top of a previous block. The finality can be expressed as $f(t) = \min(3t/(2nb), 1)$, where $f(t)$ is the probability of finality at time t, t is the time in seconds, n is the number of nodes, and b is the block time. t is discrete and can only be a multiple of b. The finality of a given block is fully reached once 2/3 of the nodes (in stake) have built a new block on top of the block to be finalized.
- *B8.3 Hybrid*. This feature includes a combination of methods such as Proof of Activity that uses two steps; A first step is based on PoW followed by a Delegated Proof of Stake (DPoS) step. During the PoW step, all stakeholder validators compete to find a solution for the header of the blocks. Once a validator has found the solution, it is broadcasted to the validators. In the DPoS phase, stakeholder validators are randomly chosen to

sign the transactions in the block. Another approach is the Proof of Space (PoSp) [106], a hybrid mechanism between PoW and PoS. PoSp is similar to PoW, except that instead of investing in computation, the user allocates memory or disk storage. With PoSp, in a first step, a verifier sends to the prover a non-trivial piece of data (e.g., typically 150 GB) that the prover has to keep. Then the prover sends a piece of data to a verifier to prove that the given amount of space has been reserved. Later, the user can engage in the PoSp proofs using minimal computation. This mechanism rewards nodes according to the amount of hard drive space they make available to the network.

- *B8.4 Byzantine Fault Tolerant/Practical Byzantine Fault Tolerant) BFT/PBFT* [66]. PBFT achieves consensus in an environment where communication is asynchronous, message delays are bounded, and at most $n/3$ of the servers are Byzantine servers. Other examples like BFT Zyzzyva [107], FaB Paxos [108], and XFS [109], provide different trade-offs on the number of Byzantine nodes which can be tolerated, as well as the efficiency of the communication.
- *B8.5-Not specified*. This feature refers to the fact that a consensus mechanism is not specified or not foreseen in the system. For instance, in the Hyperledge Fabric, even if pluggable, the consensus is not natively included in the system. The architecture implements Kafka-based ordering services intending only to be fault-tolerant.

B9-Type. This property allows qualifying the kind of BC adopted in the application.

- *B9.1 Custom*. Several solutions are based on modified versions of existing software or entirely new approaches. Custom versions are justified by the intent of improving the drawbacks of existing solutions, mainly on performance.
- *B9.2 Bitcoin*. This denotes a DTMS/DRMS based on Bitcoin technology.
- *B9.3 Ethereum*. This denotes a DTMS/DRMS based on Ethereum technology.
- *B9.4 Others*. In this feature, other emerging solutions such as Multichain, Hyperledge Fabric, and BigChainDB, are included. Their usage is uncommon since they are substantially very recent.

V. SURVEYED PAPERS

A. SCIENTIFIC PAPERS

Domains in which DTMS/DRMS have used BC are very diverse, as shown in Table 1. The table reports the application fields as well as the intended improvements derived by the use of BC, as stated in the selected papers. These improvements seem to converge on privacy, security, and information trustworthiness.

In [110], a BC-based DTMS/DRMS is used to implement security in the DNS, overcoming the limits of the Domain Name System Security Extensions (DNSSEC) protocol that is highly centralized. The authors proposed a solution based on Namecoin [126] called Flatcoin, where PoW is used. The objective is to reduce the number of built-in CA certificates to the minimum required level defined by the end-user. The proposed Flatcoin wallet key pair is used to generate a dependent X.509 certificate for webmail (DKIM or S/MIME) and client authentication. Then each user becomes a Trusted Third Party (TTP), and the customers will buy the certificate from the TTP with the highest reputation score. The gross amount of transaction fees paid serves as a reputation score metric. In the field of services, a BC-based DTMS/DRMS solution has been introduced in the consumer-producer environment [111]. The system adopts the voucher as a transaction linked to a past payment. The voucher contains an amount of money (voting fee) equal to a percentage of the payment, and optionally, it can contain an additional amount as an incentive. Both parties must digitally sign it to finalize a transaction.

TABLE 1. Field of application and intended improvement.

Paper	Field of application	Intended improvement
[110]	Domain Name System (DNS)	No central Certification Authority (CA)
[111]	Services	
[49], [112]	P2P	Withstand attacks, and security
[113], [114]	e-Commerce	Privacy, trustless system, and security
[115]	Education	Certification
[116], [117], [118]	Crowd-sourcing/sensing	Privacy and security
[68], [119], [120]	IoT/Sensor Network/Edge computing	Privacy and security
[121], [122]	VANET	Legal evidence, privacy, and security
[123]	Robotic	Information trustworthiness
[101]	Autonomous Systems	Information trustworthiness
[124], [125]	Generic Multiagents	Authorization, privacy, and security

BC-based DTMS/DRMS has also been introduced in P2P networks. In [49], the objective is to solve the issue of quantifying reputation by removing the personal opinion from the transaction in a P2P network. In the system, a positive transaction is classified as a transaction in which the user received the requested file. To reduce malicious transactions on the network, they propose a PoS approach, where a user with a low, or no reputation puts as stake a small amount of currency (bitcoins). Reputation is saved on the BC, and the client calculates the reputation score based on its parameters and only over a short period because of calculation accuracy [127]. Another work focusing on resource sharing in P2P networks is presented in [112], where a multi-level reputation system for a Cluster Of Non-Dedicated Interoperating Kernels (Clondike) [128] is proposed. The architecture allows users to contribute with the computing power of their machines and, in turn, consume the power of the other machines for their computations [129]. In order to regulate a fair usage of resources among all nodes of an inter-organization cluster, a multi-level reputation scoring system based on rewards (Kudos), is defined. BC is used for logging node activities, allowing any single node to calculate the reputation of a given node to identify and eliminate nodes that tend to overuse resources of the whole cluster and do not contribute by their computation resources or contribute by false results.

Other examples of BC-based DTMS/DRMS are proposed in the e-commerce domain [113], [114]. In [114], the authors propose a solution to lower the overhead for the processing

of transactions. The customer retrieves the Service Provider (SP) reputation to decide whether to engage in a transaction. Once the transaction is completed, the customer receives a token from the SP based on the amount available on its account (to prevent ballot-stung attacks). Then, the customer broadcasts a message containing the address of the SP, the token, and the rating of the transaction. The message also contains (optionally) a written review, a signature on this information, as well as a pointer to the last review concerning the same service provider. The pointer enables any participant to compute the reputation much faster since it does not need to retrieve the entire reputation history. In [113], instead, the aim is to analyze the underlying transaction network structure and build a history of transaction outcomes to eliminate the need for third parties. In the system, users cannot directly rate or influence the score of others. The reputation system is composed of a series of SCs executed by developers who are running Bitcoin nodes. In order to run these programs, the developer must also have another BC like Counterparty or Ethereum installed on their Bitcoin node. Each user runs his proprietary reputation algorithm to calculate its subjective score and decide whether to engage in the transaction. If the transaction occurs, the user runs the transaction contract to store the outcome, his numerical rating for the relative ''success'' of the transaction, in the Ethereum or Counterparty BCs.

In [115], a BC-based DTMS/DRMS is implemented in the educational field to store records of achievement and credit, such as degree certificates. The authors propose a permanent distributed record of intellectual effort and the associated reputation based on BC, to democratize educational reputation beyond the academic community. Each participating organization and intellectual worker receives an initial amount of educational reputation currency (Kudos), based on some existing metrics (e.g., h-index for academics). An institution allocates some of its Kudos to staff whose reputation it wishes to promote. Then, any participant can make a reputation-based transaction. The amount of Kudos represents the value of the work or idea.

During the literature analysis, we have identified three works [116]–[118] in the crowdsourcing field. In particular, in [116], the authors analyze three essential aspects of crowdsourcing/sensing: user participation, data sensing quality, and user anonymity. They introduce PaySense, a general framework that promote user participation and provides a mechanism to validate the quality of collected data based on the users' reputation. The approach is to adopt bitcoins as a reputation annotation system, unifying the concepts of reward and reputation in a unique value. In this respect, the balance in a specific Bitcoin address represents both the total awarded bitcoins for the sensing tasks reported with such Bitcoin address and the reputation obtained for the tasks. Another application where a crowd of workers can solve a requester's task without relying on any third trusted institution is proposed in [118]. Users should first register before starting the crowdsourcing, and a default reputation value is assigned. The value is changed upon the worker's behavior. In order to get satisfactory results, the requester only allows qualified workers who reach a minimum reputation value to receive the task. In [117], a Proof-of-Trust (PoT) consensus protocol, for enhancing accountability in crowdsourcing services, is provided. The authors present a novel approach that separates the transaction validation and block recording in two different groups. The goal is to achieve a better trade-off between centralization/decentralization and security/fairness. The work proposes a hybrid BC solution that utilizes a permissioned BC as the underlying deployment architecture, while the transaction validation of the consensus protocol is performed through an open, public network environment, which exhibits the fairness and impartiality properties of a public BC.

BC-based DTMS/DRMS have also been used in IoT/sensors networks to guarantee security and consistency, as presented in [68], [119], [120]. Specifically, in [119], BC is used as generic storage to manage trust and authentication for decentralized sensor networks. The approach implements a properly sized payload for storing essential security and trust information in a Bitcoin-based BC. The payload contained in the BC is used as an indication of a node's behavior over time. Reputation and trust are derived by the event analysis saved in the BC. The lack of trust between devices in IoT, where there is no common root of trust, is addressed in [120]. The objective is to allow anyone to consume services by providing a public obligation for fulfilling the terms of use as specified by the SP. The authors have leveraged the trust that users already have with their mobile operators to provide a complete path of trust between any customer and the SP. In this regard, BC is used to create an obligation chain, which is a new platform for a distributed credit-like system. This credit system has a built-in reputation mechanism that allows peers to decide whether or not to accept obligations based on the credit history of a consumer.

In vehicular ad hoc networks (VANETs) environments, the works in [121] and [122] have addressed the high variability in the topology that makes the timely evaluation of the trustworthiness of the received messages challenging. Here a BC-based DTMS/DRMS has been used by vehicles to calculate the credibility of each message about an event received from other vehicles. Then all the messages received are aggregated based on the event they refer to, to decide if the event occurred or not. Once the in-vehicle assessment is performed, a vehicle receives a reputation score based on its behavior.

The issue of reliability in knowledge and information sharing is also tangible in the swarm robotics field, where Byzantine robots may affect the quality of the generated knowledge. A BC-based DRMS is proposed in [123] to detect robots performing arbitrarily faulty or malicious behavior. Each robot represents a node in an Ethereum private network. If the distance between any two robots is smaller than 50 cm, the robots can exchange their BC information (blocks and transactions). The absolute difference between the value sent by the robot and the mean of all sent values of all robots is

calculated and stored. This difference is then used to update the robot's reputation value.

In the context of Autonomous Systems, an example of permissioned BC-based DTMS/DRMS to support the selection of an SP among possible providers is proposed in [101]. The approach is to characterize the quality of an SP based on the conformance of its network performance with Service Level Agreement (SLA)s of interconnection agreements. This is verified by SCs to compute an SLA score for each Autonomous System (AS) and to identify false testimony about forwarding performance.

A permissioned BC-based DTMS/DRMS is also used in [124] to manage trust and reputation in a distributed multi-agent system employed in safety- and information-critical domains. To attain a trusted environment, the authors propose a system that allows the agents to interact with each other and enables tracking how their reputation changes after every interaction. Agents' reputations are computed transparently using SCs. The BC stores reputation values, as well as services and their evaluations, to ensure trustworthy interactions between the agents. Finally, in [125], a BC-based authorization and access control is presented. The authors propose a hybrid design for scalable and secure trust management on a global scale so that access delegations and trust assessments could be exchanged through a BC. The global layer represents the backbone of the system. It consists of miners maintaining a public BC, and it can be instantiated upon existing BC (e.g., Ethereum). Miners are incentivized to invest computational power with fees paid for each operation on the BC. The decision on whether or not to request or delegate access to the resources is based on experience-derived reputation. Therefore, the authors propose incorporating ratings by the interacting parties, as part of the access delegation process that is achieved through the BC.

B. BUSINESS INITIATIVES

Several websites are presenting new ICOs and startups, claiming that they are going to deliver BC-based reputation systems. Unfortunately, most of them are just a sort of declaration of intent with few or no business or technical details attached. In this section, we focus only on those whose technical documentation has been recovered, basically from white papers.

MONETA [130] is presented as a decentralized reputation system with the aim of allowing both parties of a transaction to avoid and solve problems such as charge-backs, damaged goods, and scams — all of which can lead to lengthy and expensive resolution processes. The trust rate of the merchant and reviews made by other clients in a merchant's website are visible to allow a client to decide if he/she wants to buy from this merchant or not. The payment is managed with Ethereum-based currency, and all the details of the purchase are saved on the BC. A built-in algorithm analyzes the quality of each transaction, assigns, and adjusts each party's trust rating according to the taken actions. The history is then recorded onto the immutable Ethereum BC.

The solution proposed by REPU.IO [131] aims at completely replacing the system of ''likes'' with the rating - both for users and for companies. The rating, subsequently, can be used as an assessment tool in various areas of the individual's life - starting with professional skills, financial stability, and ending with all possible aspects of behavior and interaction with the public.

UTEMIS [132] is a Latin American project whose mission is to find new clients and suppliers and evaluate them by their reputation. Every time that a transaction is concluded in the UTEMIS platform, both buyers and sellers must rate the experience. UTEMIS uses Ethereum BC to store both economic and reputational transaction results.

DREP Chain [133] proposes a new scale-out architecture with a mutually independent governance mechanism for the two-layer structure that quantifies, monetizes, and aggregates the reputation value of users across different platforms (e.g., e-commerce, gaming, social networks). The reputation protocol is designed to serve the reputation-based assets and currencies running both inside the platform ecosystem and across platforms. Reputation quantification is mainly based on the behaviors of all participants on the Internet, including publishing content, commenting, rating, voting, sharing, and trading. By introducing an economic incentive mechanism centered on the DREP token, all participants are encouraged to value and maintain their reputation.

DREAM is presented in [134] as an ''Identity and Reputation'' system management, whose goals are to manage the authenticity of freelancers' identities and support reputation interoperability. DREAM tackles these challenges by remaining BC agnostic and creating a solution where participants can easily carry their reputations from DREAM to other platforms. The nature of the DREAM tokens is twofold: there are utility tokens that can be used to buy premium services and hire talent, and there are also reward tokens that encourage community members to use and to grow the platform.

In Atonomi [135], what is tokenized is the identity and reputation of devices, leveraging Ethereum. Key participants such as device manufacturers, distributors, device owners, and auditors, receive Atonomi tokens for participating in the Atonomi trust environment. A smart contract issues tokens to auditors and manufacturers of the reporting devices, according to the parameters set by the manufacturers. The token is the measure of the trustworthiness in the Atonomi environment.

Enigma [136] aims at building a privacy layer for the decentralized web. Staking is used to discourage malicious behavior by having individuals running nodes stake a value that they lose when they act harmfully (either intentionally, or unintentionally). Staking history can be a valuable source of on-chain reputation. Enigma aims at bringing together operational reputation from different sources exploring reputational building blocks around payment reliability, performance reliability, and other types of transaction histories.

TABLE 2. Scientific papers - Blockchain feature assessment.

Scientific papers	B1		B2		B3		B4			B5		B6		B7		B8					B9			
	B1.1	B1.2	B2.1	B2.2	B3.1	B3.2	B4.1	B4.2	B4.3	B5.1	B5.2	B6.1	B6.2	B7.1	B7.2	B8.1	B8.2	B8.3	B8.4	B8.5	B9.1	B9.2	B9.3	B9.4
[110] 2013	*		*			*		*		*		*			*	*						*		
[111] 2015	*		*			*				*		*		*		*						*		
[49] 2015	*			*		*		*		*			*	*			*				*			
[114] 2016	*		*		*				*	*		*			*		*						*	
[115] 2016	*		*		*				*		*		*		*		*						*	
[116] 2016	*		*			*	*			*		*			*	*						*		
[119] 2017	*			*		*		*		*			*	*		*					*			
[112] 2017		*		*		*		*		*			*		*				*					*
[121] 2018		*		*		*		*		*			*		*			*			*			
[113] 2018	*		*		*				*	*		*		*		*							*	
[123] 2018		*	*		*				*	*		*		*		*							*	
[101] 2018		*		*	*				*	*			*	*						*	*			
[120] 2018	*		*			*		*		*			*	*		*					*			
[117] 2018		*		*		*	*				*		*	*				*			*			
[124] 2018		*		*	*				*	*			*	*						*				*
[125] 2018	*			*	*				*		*	*		*		*							*	
[122] 2018		*		*		*		*		*			*	*		*					*			
[118] 2018	*		*		*				*	*		*		*		*							*	
[68] 2018		*		*	*				*		*		*	*						*				*

TABLE 3. Scientific papers - DTMS/DRMS feature assessment.

Scientific papers	A1				A2		A3			A4			A5				A6		A7			A8			A9		A10		A11	
	A1.1	A1.2	A1.3	A1.4	A2.1	A2.2	A3.1	A3.2	A3.3	A4.1	A4.2	A4.3	A5.1	A5.2	A5.3	A5.4	A6.1	A6.2	A7.1	A7.2	A7.3	A8.1	A8.2	A8.3	A9.1	A9.2	A10.1	A10.2	A11.1	A11.2
[110] 2013			*			*		*		*					*		*				*	*			*		*		*	
[111] 2015	*					*	*			*					*		*		*			*			*		*		*	
[49] 2015	*					*			*	*				*				*			*	*			*			*		*
[114] 2016	*					*	*			*			*				*		*			*			*		*		*	
[115] 2016	*				*		*			*				*			*		*			*			*		*		*	
[116] 2016	*					*		*		*					*		*		*			*			*		*		*	
[119] 2017				*	*		*				*		*				*			*			*			*		*		*
[112] 2017	*				*			*		*					*			*	*			*				*		*		*
[121] 2018	*					*		*		*				*			*		*				*			*		*		*
[113] 2018	*					*	*			*				*			*		*			*				*		*		*
[123] 2018		*				*	*					*	*				*		*			*			*		*		*	
[101] 2018	*					*		*		*			*				*		*				*			*		*		*
[120] 2018			*			*	*			*			*					*	*			*			*			*		*
[117] 2018			*			*		*			*			*				*	*					*		*	*		*	
[124] 2018	*					*		*		*			*				*		*				*			*		*	*	
[125] 2018	*					*	*				*		*				*		*					*	*				*	
[122] 2018			*			*		*		*			*				*		*			*				*	*			*
[118] 2018	*				*			*		*			*				*		*				*			*	*		*	
[68] 2018				*		*		*		*				*				*			*		*			*		*		*

TABLE 4. Business initiatives - Blockchain feature assessment.

Business applications	B1		B2		B3		B4			B5		B6		B7		B8					B9			
	B1.1	B1.2	B2.1	B2.2	B3.1	B3.2	B4.1	B4.2	B4.3	B5.1	B5.2	B6.1	B6.2	B7.1	B7.2	B8.1	B8.2	B8.3	B8.4	B8.5	B9.1	B9.2	B9.3	B9.4
[130] 2017	*		*		*				*		*	*			*	*							*	
[131] 2017	*		*		*				*		*	*			*	*							*	
[137] 2017	*		*		*				*		*	*			*				*		*			
[132] 2018	*		*		*				*		*	*			*	*							*	
[135] 2018	*		*		*				*			*			*	*							*	
[133] 2018	*		*		*				*		*	*			*					*	*			
[136] 2018		*		*	*				*		*	*			*					*	*			

TABLE 5. Business initiatives - DTMS/DRMS feature assessment.

Scientific papers	A1				A2		A3			A4			A5				A6		A7			A8			A9		A10		A11	
	A1.1	A1.2	A1.3	A1.4	A2.1	A2.2	A3.1	A3.2	A3.3	A4.1	A4.2	A4.3	A5.1	A5.2	A5.3	A5.4	A6.1	A6.2	A7.1	A7.2	A7.3	A8.1	A8.2	A8.3	A9.1	A9.2	A10.1	A10.2	A11.1	A11.2
[130] 2017	*					*		*		*			*				*		*			*			*			*	*	
[131] 2017	*					*		*		*			*				*		*			*			*		*		*	
[137] 2017	*					*		*		*			*				*		*			*			*		*		*	
[132] 2018	*					*		*		*			*				*		*			*			*			*	*	
[135] 2018				*		*		*		*			*				*		*				*			*				*
[133] 2018	*					*	*			*			*				*		*				*		*		*		*	
[136] 2018	*					*		*		*			*				*		*				*		*			*	*	

Finally, Bitconch chain [137] proposes an innovative Proof of Reputation consensus algorithm, which offers a new solution that leverages BC technology to maintain both high throughput and decentralization.

VI. LITERATURE ASSESSMENT

According to Step 3 of the methodology (Section III), the literature is analyzed, and the features related to BC and DTMS/DRMS of each selected article have been reported in the lattices forming the formal context (Tables 2, 3, 4, and 5). We applied some fundamental statistical analysis to identify significant characteristics and trends.

A. TRENDS ANALYSIS

According to the feature analysis, several properties show a temporal-based trend that is important to forecast the future next steps of the BC adoption in DTMS/DRMS. The most

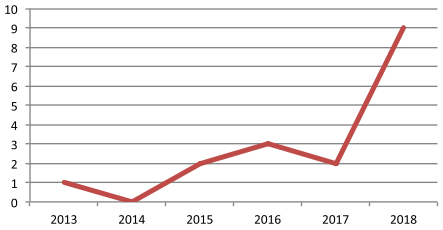

FIGURE 4. Publication trend (in number of papers).

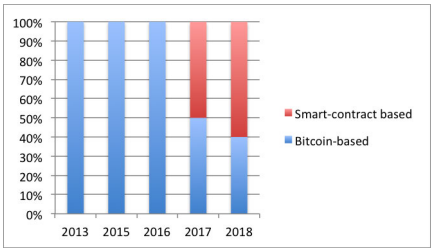

FIGURE 5. Smart contract usage trend.

evident one is related to the growing interest in the use of BC technology in DTMS/DRMS (Fig. 4).

The second relevant trend is related to the progressive increase in using the on-chain Business Logic approach (B3.1) exploiting BCs with smart contracts like Ethereum (Fig. 5).

This trend is very similar to what is exhibited by *B*1 and *B*2 properties, where there is a general shift from public and permissionless towards private and permissioned BC. Properties like *B*5.1, *B*7.1, and *B*8.1 do not seem to be affected by these changes and exhibit stronger stability. This is particularly highlighted with the FCA analysis. By applying the FCA-based transformation, the formal context is represented as a lattice where each node represents a concept according to the provided definition. In order to identify the most invariant properties, we defined 50% as a threshold for the size of the support *s* of a single property to be considered invariant in the analysis.

The result of the analysis is depicted in Fig. 6. Objects are represented in the lower part of the figure, while properties with a percentage of objects aggregated and filtered according to *s* are represented in the upper part.

From the FCA analysis, the following properties emerged as more stable or invariant (Fig. 7).

1) B1.1 - OPENNESS - PUBLIC

According to the results of the analysis, the majority of BC-based DTMS/DRMS adopted a public BC like Bitcoin. In particular, in [110], the authors presented Flatcoin (based on Namecoin), the first fork of the Bitcoin, and still, one of the most innovative "altcoins." The system reveals the transaction information and guarantees the integrity of the transaction database that is managed in a P2P manner. In [49], the authors introduce a custom BC with a PoS approach. Furthermore, they exploit the merge mining technique with the Bitcoin network to address the

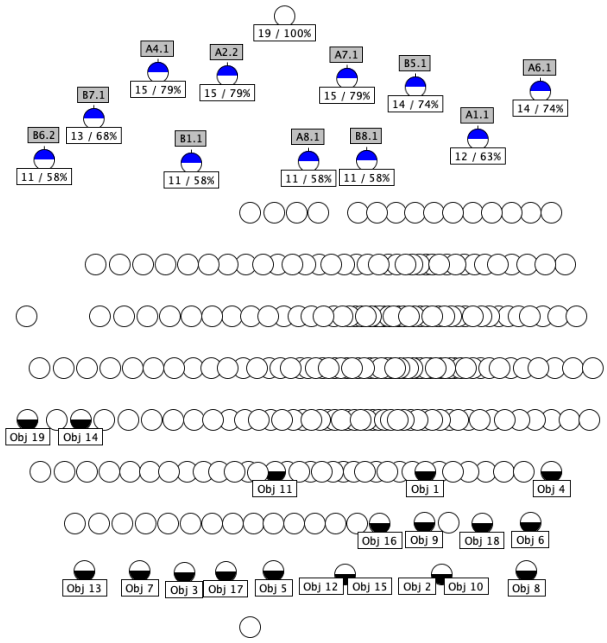

FIGURE 6. FCA analysis results.

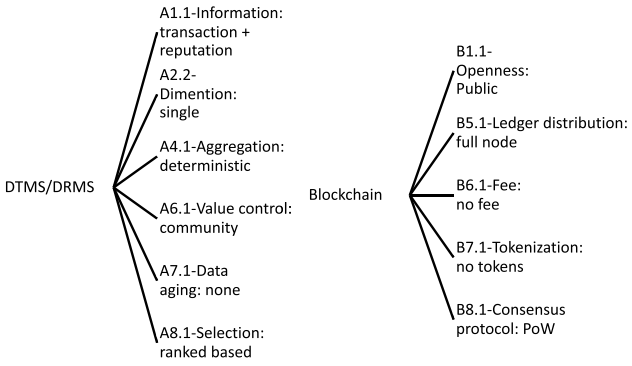

FIGURE 7. The most stable properties.

cold-start problem. The DTMS solution proposed in [114] for the e-commerce domain is based on a trustless scenario and assumes the presence of Turing-complete scripting language. The authors proposed the use of the public Ethereum, defining a robust protocol able to cope with attacks such as bad-mouthing and ballot-stuffing. In [115], BC is seen as a publicly-accessible distributed digital record where an educational organization can store records of achievements and credits such as degree certificates. The use of public BC like Bitcoin to support the DTMS/DRMS mechanism has also been proposed in [111], [113], and [116]. In [119], the authors proposed to use BC as a general decentralized, secured data storage structure exploiting a public BC, even if only authenticated nodes can mine new blocks, and only if they have not issued a payload to be included in the block.

In [120], the obligation chain, a decentralized reputational model for IoT based on credit, is proposed. In the obligation chain, IoT devices can perform a transaction using credit, and their ability to pay back their credit adds to their

reputation [53]. Thus, all transactions publicly recorded are part of a distributed reputation platform.

In [125], trust management is used for access delegation in IoT, implementing a global distributed ledger. While, in [118], a public BC-based decentralized framework for crowdsourcing, named CrowdBC, is proposed. CrowdBC guarantees privacy by allowing users to register without a real identity.

2) B5.1 LEDGER DISTRIBUTION - FULL NODE

This feature identifies those solutions that require for the participants to implement a full node such as [101], [120], [123], and [124]. In [111], each actor in the network (producer and consumer) runs a full node, as in a Bitcoin network. Also, in [116] and [113], the Bitcoin network is used as a reputation system. Each participant should have a copy of the entire ledger. In [49], the node has a full ledger. However, the ledger does not store all the information. It used the ''friend peer reputation'' model, saving the history of the interaction at the local level. Although not clearly stated, in [114], each participant should have a full node implementation by the fact that they are oriented to implement their solution with Ethereum. In [119], the management of a full ledger is also required, but it is managed by authorized participants entitled to mine new blocks.

In [112], the BigChainDB technology [138] is used. BigChainDB is a distributed database with some BC features like immutability and owner-controlled assets. In BigChainDB, all the nodes have a copy of the ledger and integrate Tendermint for inter-node networking. The works that address the VANET topic like [121] and [122] assign to the road side units (RSUs) in a VANET the role of hosting a full copy of the ledger, and in turn, to perform a PoW for mining the block. In [118], in order to reduce the size of the ledger, the DTMS/DRMS divides the application into three layers: the application layer, the BC layer, and the storage layer. The application and BC layers lie in the logic plane and the storage layer lies in the data plane. Thus, users do not need to trust the data saved in the data layer, and they can verify the integrity and authenticity of data in the logic layer.

3) B6.1- FEE - NO FEE

Most of the analyzed papers do not discuss the fee property. It is worth to notice that none of these papers has clearly stated that their proposed solution avoids a fee mechanism. This evaluation has been done indirectly because a fee mechanism is not mentioned/addressed or because the underlying technology (e.g., Fabric) usually does not include a fee for the transaction.

In [122], the authors use a Bitcoin-like BC that includes a fee mechanism in the system, and in [121], a combination of PoW and PoS may suggest that a sort of payment should be defined. The works presented in [68], [101], [117], [120], and [124] do not address the fee property. In general, even if a fee system is not explicitly excluded, at the scientific level, its implementation or definition does not seem to deserve attention. On the other hand, in the business-related applications, the scenario is the opposite, where all the analyzed solutions consider the fee system as a core part of the value proposition.

4) B7.1 - TOKENIZATION - NO TOKENS

The majority of the analyzed solutions avoid using tokens in DTMS/DRMS. In [113], the reputation in a transaction network, such as Bitcoin, is calculated using net flow rate convergence, so that it is not necessary to have reputation tokenization. Similarly, in [68], [101], [117], [119], [120], [122]–[124], and [125], the adopted solutions do not include the use of tokens, although the fields of application are very diverse. In [118], despite the fact that Ethereum is used and that ERC20 is natively available, the reputation mechanism is not tokenized.

5) B8.1 CONSENSUS PROTOCOL - POW

Although several alternatives are under development and testing (e.g., PoS), the general preference remains for the PoW approach. In particular, in [110], the authors propose to use PoW to timestamp a decentralized P2P database of events for the DNS. Similarly, in [111], the system is based on Bitcoin, and hence, the consensus protocol is intrinsically the PoW. The Bitcoin network is also directly used in [116], while customization of the Bitcoin infrastructure is adopted in [119]. PoW is also part of the solutions provided in [118], [120], [123], and [125], where the aspects of scalability and performance are not considered.

6) A1.1 - INFORMATION - TRANSACTION + REPUTATION

This property highlights the tendency of having in the DTMS/DRMS both transactions and the reputations of the agents. This approach makes the system trustworthy and consistent, reducing the points of failure and vulnerability. Decoupling transaction and reputation management requires an extra effort in terms of system integration, security, and validity check.

In [111], both money transactions and vouchers are managed at the same time. The consumer requests a service and performs the payment, and if satisfied, the user decides whether or not to accept the incentive in the voucher, leaving positive feedback. In [49], the authors propose to manage reputation about the transactions that occurred within the BC, calculating both objective and subjective reputations from peers. Similarly, in [114], the BC stores not only the transaction, but also, the associated rating (including a review about it), the identifier of the SP, the token used for the review, and the token's signature. In the solution presented in [115], the BC registers scholarly productions and supports Kudos transfer from a participant to one or more of such registrations. Other people might then transfer some of their reputational credit to the author, to boost the reputation of that person's artifact or idea. Since in [116], bitcoins represent both reward and reputation, the authors designed a system where, before sending sensed data to the server, crowd sensors must first obtain a Bitcoin address certificate from

the Address Certification Authority (ACA). The address will enable them to receive rewards and reputation updates in the form of Bitcoin payments for the sensing tasks they perform. In [119] instead, six payloads corresponding to the events managed by the system: Miner approval, Credentials, Renew, Blame, Ban, and Revoke have been defined for the BC. For each payload type, the system defines events and associated reputation factors. In [101], BC is used to store network measurements (considered as a transaction) to verify whether the forwarding performance of an AS is in conformance with SLAs of interconnection agreements. Smart contracts compute an SLA score for each AS and identify if any of them has provided false testimony about forwarding performance. In the solution presented in [124], the information stored in the BC is complex and includes a) the information about the service(s) provided by the agents, b) the information about the interactions that took place in the community, and c) the detailed evaluation from both service's demander and executor. Since the solution is based on Hyperledge, which natively uses levelDB, the data are modeled as a database and then implemented in the chaincode. Finally, in [118], the authors managed the DTMS/DRMS information using SCs. They identified three types of SCs: user register contract, user summary contract, and requester-worker relationship contract. Each of them has its own data model. Information is divided into basic and detailed. The basic information which contains the name, address, and type is saved in the global user register contract. The detailed information, including the user's profile, expertise, reputation, and task list, is saved in the user summary contract. When the task terminates, the value of reputation and expertise are automatically updated.

7) A2.2 - DIMENSION - SINGLE

The DTMS/DRMS that implement BC tend to focus on a single aspect among all possible behavior on which an agent is deemed reliable. Thus, it is not possible to calculate trust in one aspect deriving the value from another aspect.

In [111], the feedback a consumer can provide refers only to the single service purchased from a service provider. A consumer that signs the voucher together with the producer validate the execution of the contract. In [110], what is evaluated is the reputation of TTPs based on their economic performance. In [49], the authors managed more than one variable contributing to reputation in a P2P reputation system. However, according to the authors, these variables cannot be classified as new dimensions but as a further qualification of a single dimension (behavior). In [114], reputation is evaluated based on the service provided in the marketplace. In [116], the reputation value quantified in bitcoins is assigned to the quality of the information provided by a sensor to the Data Collection Server. In [121], the value expresses the reputation of a vehicle in broadcasting reliable messages in a VANET as in [113] where reputation expresses the reliability of the actor performing the transaction.

Similarly, in [101], the reputation system for the Autonomous System is devoted to avoiding the selection of a provider that fails to guarantee the agreed SLA without any other derivable implication. The solution developed in [124] treats a single dimension reputation calculating the overall rating for the agent as well as a task-dependent reputation value. In [122], the reputation expresses only the level of trustworthiness of a vehicle in broadcasting reliable messages in the VANET. Other works like [68], [117], [120], [123], and [125] are using the same approach.

8) A4.1 - AGGREGATION - DETERMINISTIC

The methods to calculate reputation in [68], [118], and [120] do not take into account probability or similar methods for calculation. In [110], the reputation is calculated by computing the amount of transaction fees put on the certificate-signing transactions. Thus, the gross income is used as a trustworthiness indicator. The reputation score is based only on the positive feedback, the trust diminishes automatically when the gross income flow fades out, and thus, the negative feedback is not needed. Since in [116], the concepts of reputation and reward are represented by bitcoins, the amount of currency in the account represents the sensor's level of trust. This is similar to the solution proposed in [115], where tokens (Kudos) are used as a method to quantify and transfer reputation. The amount of trust is calculated by counting the number of Kudos associated with a participant or an intellectual product registered in the BC. The Bitcoin-based computation of the reputation and trust is also adopted in [116]. The counting aggregation method is also used in [49] and [111], where for instance, the total reputation is equal to the sum of voting fees. In order to compute the reputation in the system proposed in [114], a customer only needs to access the last block containing a review about the SP, whose reputation it seeks. It is then sufficient to follow the pointers in order to retrieve all the reviews about this SP. For each review, the prospective customer can choose any aggregation functions such as mean or median. In the work presented in [112], each node could have its specific reputation calculation strategy. However, they should be based on Kudos. In order to take into consideration multiple types of behavior, the authors introduce multi-level scoring with different Kudos values. The calculation of reputation in [121] is a bit sophisticated and is divided into two phases. In the first phase, each vehicle receives a message from other vehicles in the VANET and calculates the aggregate credibility of an event using a Bayesian Inference. The ratings are periodically uploaded to the RSU hosting the BC. In the second phase, the RSU algebraically calculates the trust value offset (if it gets different ratings) about a specific message and performs a PoW among the other RSU to mine the block.

The computational method to estimate the reputation in [113] is based on the net flow convergence method. The inputs are the transaction data for the P2P network, and the outputs are the rates of convergence of each node. The method to calculate reputation in [101] is defined by a pre-agreed and publicly known scoring function. The system developed in [124] provides two values: a) an overall reputation value

rating the average agent's reputation; b) a task-specific value of a given service and role (demander/ executor). Finally, in [122], the reward (or punishment) for a vehicle is calculated considering the level of alert of the broadcasted messages, the density of the vehicles, and the sequence of the senders. These factors are algebraically combined with a reward or penalty coefficient to obtain the final score.

9) A6.1 VALUE CONTROL - COMMUNITY

The value of reputation and trust can be defined subjectively and used as it is by the agent, or it can be negotiated and mitigated among the other agents in the network, in order to reach a consensus on an objective value. The majority of the approaches implement a community-based control on the reputation value as defined in [118], [119], [123], [124], and [125]. In particular, in [110], the reputation value is derived by the income accumulated by TTPs and their transaction history that is saved on the BC. Because it is Bitcoin-based, the reputation score in [111] is controlled by the community since all the participants host a replica of the ledger. Similarly, in [114], all the ratings and the transactions generated by the participants are saved in the BC. This allows a community-based control on the provided rating. The community also manages the reputation and trust value in [115], since the final score is based on the aggregation of the public votes (Kudos) assigned by the network participants, instead of a subjective evaluation of a single user. This is similar to what is proposed in [116], where the amount of bitcoins of a participant represents its reputation. This value is visible and accepted by all the participants in the system. Both subjective and objective perspectives of reputation are considered in [113]. The system can store privately calculated reputations for counter-parties, or can make the result of the reputation calculation publicly available on the Ethereum or Counterparty BC using the net flow convergence algorithm. In [101], the algorithm to calculate the score should be publicly known and agreed upon among the participants. Moreover, the results are written in the BC. However, to preserve the privacy of the participants and allow other parties to verify any results, the authors adopted the order-preserving encryption [139]. Finally, the solution proposed in [122] uses two BCs managed by all RSUs composing the VANET, to prevent vehicles from broadcasting forged messages while simultaneously protecting the privacy of each vehicle against tracking attacks. Similarly, in [121], the reputation is calculated by RSUs and saved in the BC, allowing a community-based control.

10) A7.1 - DATA AGING - NONE

Managing data aging in a DTMS/DRMS is quite complex and includes the definition of thresholds for the time window of value validity and a degradation function. However, even if there are valid reasons to implement such a mechanism, the majority of the surveyed works decided not to adopt it. In the works presented in [101], [111], [113], [117], [118], [120]–[124], and [125], no aging mechanism is considered. This means that the dynamics of the reputation depend exclusively on the agent's behavior. Thus, past transactions weight as much as the recent ones in the reputation calculation. In [114], the problem of data aging has been mentioned only in terms of system scalability. They argue that one can choose to ignore ratings that are older than a given age. Since the reputation calculation method in the proposed solution requires traversing the BC following the review thread associated with an SP, a customer could download fewer blocks. Although all the records are timestamped to provide a trusted and timed record of the added data, in [115], the data aging technique is not defined. Thus, the tokens used to quantify the reputation do not have an associated expiration date similar to the Kudos in [112] that remains valid permanently. Finally, the work in [116] does not clearly define a mechanism to manage transaction and reputation aging. However, it uses an ACA. The ACA has two goals: ensuring that participants (crowds sensors) do not use more than one Bitcoin address at a given time and that Bitcoin addresses are renewed periodically given the expiration date of the certificate, which effectively limits the validity of the Bitcoin address itself. Although Bitcoin addresses do not expire, the authors apply the concept of expiration to the certificate issued by the ACA. When a certificate address expires, such an address cannot be used in the reputation system, but it is still a valid and usable standard Bitcoin address.

11) A8.1 - SELECTION - RANKED BASED

Supporting the decision of an agent to enter into a transaction with another agent is the core business of a DTMS/DRMS. Thus, the method adopted for partner selection is crucial. The majority of the analyzed works have implemented a ranked based approach. This means that the agent will start a transaction only with partners that occupy the higher level in the ranking, as implemented in [116], [120], and [123].

B. IMPLICATIONS ANALYSIS

Another interesting analysis is performed in order to identify what are the BC features that can satisfy the DTMS/DRMS requirements. The implication between two attributes $B_1 \implies B_2$ holds if any object that satisfies all the elements of B_1 also satisfies all elements of B_2. From the Duquenne-Guigues-Basis [86] for implications, one can derive all implications valid in a formal context using the Armstrong rules [140]. We analyzed and reported here only the implications that have support s above 50% and confidence c above 70%. The resulting 67 implications have been filtered in order to select only those that satisfied the following implication: $B \implies A$, to highlight the properties of the BC that satisfy the DTMS/DRMS features. A second refinement has been applied verifying if the implication was accidental or could imply a sort of causal nexus. In Table 6, the resultant implications are reported. It is worth to remark that because of the limited number of papers identified in the domain, the results should be considered preliminary. A more precise understanding would emerge

TABLE 6. Implications analysis $B \implies A$.

n	Property B	Support	$B \implies A$ Conf.	Property A	Support
1	B7.1	13	85%	A2.2	11
2	B8.1	11	91%	A6.1	10
3	B8.1	11	82%	A7.1	19
4	B6.2	11	82%	A4.1	9
5	B1.1	11	82%	A6.1	9

as soon as the technology, and the implementations become more mature and widespread.

A possible explanation of the first implication B7.1 (Tokenization - no tokens) implies A2.2 (Dimension - single) can be given by the fact that the evaluation of a single behavior in a given context does not imply or require any specific form of trade. This means that BCs that allow avoiding token management are going to be preferred. This is coherent with the shift from a public, permissionless, and off-chain business logic towards private, permissioned, and on-chain approaches. In these approaches, BC is used as a reliable and transparent platform to calculate and preserve trust and reputation values to be used for decision making rather than to trade with them in an open market.

The implication number 2 reveals a relation between the consensus method based on PoW (B8.1) and the reputation and trust values controlled and validated by a community (A6.1). This implication may suggest that the consensus mechanism should not be considered a mere technical feature of the BC separated from the business logic of the application. It embeds both technical and organizational aspects. Thus, the selection of the appropriate consensus mechanism should also be driven by the design of the application as well as the technical concerns (e.g., performance, vulnerability).

In the third implication, the relation is between the PoW and the absence of a data aging mechanism. At first, the two properties seem to be independent. However, the presence of such an implication may be justified as follows. The management of the aging of the reputation value may require the presence of a maintenance process that continuously verifies the results of past transactions. Then this process should apply a degradation algorithm to the stored values obtaining a new result that needs to be stored in the ledger in a new transaction. This maintenance process, even if it generates a new transaction, does not require the presence of PoW because it is not a transaction among agents. Thus, aging seems to be challenging to implement in BC, especially in systems using PoW.

The implication number 4 notes the presence of a relation between the existence of a fee mechanism in the system and the deterministic method to compute the reputation value. This implication is interesting since it may reveal the presence of a psychological bias of the proposers rather than a genuinely technical need. The adoption of a deterministic solution for the reputation value computation is intrinsically more straightforward and easy to be managed and verified in comparison to other solutions. We can imagine that, as soon as the solution starts to confront the market, the need to reduce the complexity emerges. This may be due to the maturity of the technology and the related skills needed to manage it properly.

The implication number 5 relates public BC with the feature of the reputation value managed by the community (A6.1). This implication is quite logical, and its presence was expected. The purpose of the public BC is to allow all actors to execute transactions and to control the transaction of others transparently. This means that the connected community is called to take the role of control and validation of the reputation value calculation and maintenance. Thus, in this case, the BC characteristic perfectly matches and supports the DTMS/DRMS feature.

Several other implications came out from the analysis, so it is worth to mention that, because of the limited number of papers available in the literature, such results are affected by a certain degree of uncertainty. Thus, the current emerged implications may change in their importance, and other more significant implications could emerge as soon as new publications become available.

VII. BUSINESS VS. SCIENTIFIC PERSPECTIVES

The results of the business based approaches reveal a substantial convergence on the following properties: B1.1, B2.1, B3.1, B4.3, B5.2, B6.1, and B7.2 related to the BC technology.

It is essential to highlight the main differences between the two domains. In particular, they differ on the following properties:

- B3: Scientific approaches are focused on off-chain business logic, while all the business-oriented approaches are adopting an on-chain solution. This result can be justified by the fact that the business applications emerged in 2017, when Ethereum was well established, and other solutions were emerging with the feature of SCs included (e.g., Hyperledge). On the contrary, scientific experimentation started in 2013 when only Bitcoin-based BC was dominant.
- B5: The commercial tools are strongly oriented to separate the nodes that host the full ledger with other kinds of participants that can interact with dedicated clients. These clients invoke SCs for the business logic and interact with the BC through them. On the contrary, the majority of scientific approaches are focused on B5.1 where all the participants host the full ledger. This difference can be attributed to the fact that in business applications, the privacy of the information is crucial. To this end, in business applications, BC tends to be used as a highly protected data storage relegating the data and computing distribution to a technical solution (e.g., the cloud) under the control of the organization or consortium.
- B6: The definition of a fee mechanism, is vital for a solution that aims to compete on the market or wants

to convince investors. On the other hand, this aspect is often neglected in scientific papers that dedicate more attention to technological issues. However, in those papers, what is explicitly claimed is that the solution is based on the public Ethereum network. Hence, the presence of a fee mechanism is implicit.

- B7: The tokenization of the reputation represents another critical point of difference. The business perspective tends to see reputation as an asset that can be traded similarly to other goods, which is why the most common token adopted is ERC20. The scientific point of view considers the reputation and trust just as a value, so the general trend is to avoid using tokens in BC-based DTMS/DRMS. However, the use of tokens in the system has several advantages. In particular, it is possible to bind the expression of votes or feedbacks to the possession of the token, that can be burned after the action. In this way, it is possible to simplify complex business processes while maintaining a high level of security and reliability.

Regarding the similarities between the two perspectives, it is worth to highlight a general preference towards the customization of BC solutions (B9.1 property). This is particularly evident in the scientific approaches, but also in the business solutions, this attitude seems to be confirmed. The customization is necessary when the features of the existing BC are not perfectly aligned to sustain the business case. In particular, there are two main customizations: The first is the adoption of new consensus protocols, moving away from PoW to other methods, in order to speed up the transaction performance. The second is the modification of the structure and dimensions of the block, in order to accommodate more information. This is particularly true for Bitcoin-based systems where data storage has several constraints. In this respect, both business and scientific approaches are experimenting with new methods and algorithms to achieve better performance while maintaining the same level of trust guaranteed by PoW. Another aspect where the two approaches are converging towards a similar trend is the use of a mechanism of incentives to reward network participants (property A10). No method seems to prevail in the business scenarios, even if there is a slight preference towards the incentives.

VIII. DISCUSSION

From the analysis carried out in the previous section, several trends emerged. The adoption of BC technologies in the DTMS/DRMS domain is in its early stages, and its evolution may follow different possible directions. However, according to the results of the analysis, we can identify some consolidated trends:

- There is a significant growth in the adoption of BC technology in DTMS/DRMS revealed by the number of publications retrieved in the year (2018) vs. the five previous years (2013-2017).
- There is a progressive shift from the Bitcoin-based technology towards solutions supporting SCs because of their flexibility and control in managing the business logic and the data storage.
- There is a tendency to shift from public and permissionless towards private and permissioned solutions. This trend could be justified by the intent of securing data privacy, and keeping the costs and the process under control.
- There is a global tendency to model a single dimension of trust. This means that it would be difficult to migrate the validity of trust calculated in one domain into another. This is also confirmed by the interoperability property that is oriented towards closed systems reducing the scope of the reputation and trust value validity to the intra-domain.
- The adoption of BC in the DTMS/DRMS domain is currently tested in different application domains that have their specific requirements. Even if the motivations to adopt BC are similar across the domains (e.g., to achieve privacy and security), the modes of implementation differ significantly, and a best practice does not seem to prevail.
- Although it is a critical element that affects the actual feasibility of a project, the performance aspect of a BC has been only partially addressed in the literature by adopting the perspective of the number of transactions per second. Unfortunately, the performance of a BC is also related to the increasing rate of updates to the ledger. However, related pruning strategies seem to be not explored yet. Another factor affecting the BC performance is related to the complexity of the contract that requires a strong conceptualization in terms of processing the business cases under analysis.
- The BC consensus protocols do not prevent some security attacks like Sybill attacks and others. It is necessary to introduce a dedicated business logic, or to consider to extend the current protocols.
- Differently from the cloud-based domain where technology is offered ''as a service'', the adoption of a BC solution at the beginning of a business-based service has an essential impact on the application's development. There are so many constraints in each approach that make the transition from a solution to another almost impossible. In order to overcome such limitations, solutions based on the ''BC as a Service'' concept are emerging. Prominent vendors like IBM, Microsoft, Oracle, and Amazon are offering cloud-based installations of BC. However, even if possible in theory, the possibility to efficiently migrate from a platform to another seems not to be well addressed.

IX. CONCLUSION

This paper provides a comprehensive survey of the recent initiatives related to the adoption of BC technology in Distributed Trust and Reputation Management systems, with a specific focus on the identification of the relevant emerging features that could represent main drivers for the next

generation of DTMS/DRMS. The analysis starts by defining the concept of trust and reputation and their role within a distributed network. We started our investigation outlining the core features of a BC-based DTMS/DRMS by grouping them into two taxonomies: a) BC related technology and b) DTMS/DRMS based systems. Moreover, the literature assessment has been conducted using the taxonomies and defining a context that has been processed by applying Formal Context Analysis. Such an approach allowed us to identify the most recurrent and stable features in the current scientific landscape and several important implications among the two sets. From the result of the analysis, we derive several recommendations that may help in saving time during the technical feasibility assessment and driving the future implementations of DTMS/DRMS. In particular, over the general recommendations of using BC because of its capability of enhancing security, privacy, and trustworthiness of the managed information, we recommend:

- Verifying the feasibility of the permissioned solutions against the business requirements.
- Adopting BC-based approaches (e.g., Hyperledger) to implement complex business logic with the use of SCs.
- Managing a single dimension of trust (single scope) per system while considering the implementation of protocols of interoperability among DTMS/DRMS in case the exchange of information about the trustworthiness of an actor is necessary.
- Carefully evaluating every performance requirement of the business application since performance is an important issue in BC. There are several factors and parameters (e.g., the number of peers composing the network, the consensus algorithm, the hardware configuration of the peers, and the complexity of the payload) that may affect the BC performance. For this reason, it is necessary to plan an optimization phase while considering that in some cases, the performance may increase by a factor of 10 or more [141].
- Starting with an in-house configuration of the BC that would be deployed on a cloud infrastructure instead of starting with a ready-to-go BC configuration provided by the cloud service providers. Even if the use of customized infrastructure speeds up the time-to-market of a BC-based application, it limits its portability, therefore, affecting the possibility of leveraging the cost reduction strategy based on the ''as-a-service'' business model.

REFERENCES

[1] N. Luhmann, ''Trust: A mechanism for the reduction of social complexity,'' in *Trust and Power*, N. Luhmann, Ed. Hoboken, NJ, USA: Wiley, 1979, pp. 4–103.
[2] D. Gambetta, ''Can we trust trust?'' in *Trust: Making and Breaking Cooperative Relations*, D. Gambetta, Ed. Oxford, U.K.: Basil Blackwell, 1990, pp. 213–237.
[3] B. Bailey, L. Gurak, and J. Konstan, ''Trust in cyberspace,'' in *Human Factors and Web Development*, N. Ratner and J. Hillsdale, Eds. Mahwah, NJ, USA: Lawrence Erlbaum Associates, 1999, pp. 311–321.
[4] M. J. Gallivan, ''Striking a balance between trust and control in a virtual organization: A content analysis of open source software case studies,'' *Inf. Syst. J.*, vol. 11, no. 4, pp. 227–304, 2001.
[5] E. Bellini, G. Bergamin, M. Messina, C. Cirinná, and R. Messuti, ''NBN: IT the Italian trusted persistent identifier infrastructure,'' *Int. J. Knowl. Learn.*, vol. 9, no. 4, pp. 347–363, 2014.
[6] M. T. Zanini, ''The new economy (NE),'' in *Trust Within Organizations of the New Economy* (International Management Studies). Wiesbaden, Germany: Deutscher Universitäts-Verlag, 2007.
[7] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, ''Reputation systems,'' *Commun. ACM*, vol. 43, no. 12, pp. 45–48, 2000.
[8] S. Ba and P. A. Pavlou, ''Evidence of the effect of trust building technology in electronic markets: Price premiums and buyer behavior,'' *MIS Quart.*, vol. 26, no. 3, pp. 243–268, 2002.
[9] G. E. Bolton, E. Katok, and A. Ockenfels, ''How effective are electronic reputation mechanisms? An Experimental investigation,'' *Manage. Sci.*, vol. 50, no. 11, pp. 1587–1602, Nov. 2004.
[10] D.-H. Park, J. Lee, and I. Han, ''The effect of on-line consumer reviews on consumer purchasing intention: The moderating role of involvement,'' *Int. J. Electron. Commerce*, vol. 11, no. 4, pp. 125–148, Jul. 2007.
[11] E. Damiani, S. D. C. di Vimercai, S. Praboschi, P. Samarati, and F. Violante, ''A reputation-based approach for choosing reliable resources in peer-to-peer networks,'' in *Proc. 9th ACM Conf. Comput. Commun. Secur.*, 2002, pp. 207–216.
[12] X. Xuy, I. Webery, M. Staplesy, L. Zhuy, J. Bosch, L. Bassz, C. Pautassox, and P. Rimba, ''A taxonomy of blockchain-based systems for architecture design,'' in *Proc. IEEE Int. Conf. Softw. Archit. (ICSA)*, Apr. 2017, pp. 243–252.
[13] E. I. Nabil and P. Claus, ''A review of distributed ledger technologies,'' in *Proc. Move Meaningful Internet Syst. (OTM) Conf.*, 2018, pp. 277–288.
[14] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, ''An overview of blockchain technology: Architecture, consensus, and future trends,'' in *Proc. IEEE Int. Congr. Big Data (BigData Congr.)*, Jun. 2017, pp. 557–564.
[15] S. M. Sarkintudu, H. H. Ibrahim, and A. B. Abdwahab, ''Taxonomy development of blockchain platforms: Information systems perspectives,'' in *Proc. AIP Conf.*, 2018, Art. no. 020130.
[16] L. Olga, D. Tobias, and S. Ali, ''From hype to reality: A taxonomy of blockchain applications,'' in *Proc. 52nd Hawaii Int. Conf. Syst. Sci. (HICSS)*, 2018, pp. 4555–4564.
[17] M. Macdonald, L. Liu-Thorrold, and R. Julien, ''The blockchain: A comparison of platforms and their uses beyond bitcoin,'' Univ. Queensland, Brisbane, QLD, Australia, Tech. Rep., 2017, doi: 10.13140/RG.2.2.23274.52164.
[18] K. Christidis and M. Devetsikiotis, ''Blockchains and smart contracts for the Internet of Things,'' *IEEE Access*, vol. 4, pp. 2292–2303, 2016.
[19] T. M. Fernández-Caramés and P. Fraga-Lamas, ''A review on the use of blockchain for the Internet of Things,'' *IEEE Access*, vol. 6, pp. 32979–33001, 2018.
[20] G. Salviotti, L. D. Rossi, and N. Abbatemarc, ''A structured framework to assess the business application landscape of blockchain technologies,'' in *Proc. 51st Hawaii Int. Conf. Syst. Sci.*, 2018, pp. 3467–3476, doi: 10.24251/HICSS.2018.440.
[21] M. Friedlmaier, A. Tumasjan, and I. Welpe, ''Disrupting industries with blockchain: The industry, venture capital funding, and regional distribution of blockchain ventures,'' *SSRN Electron. J.*, to be published.
[22] A. Jøsang, R. Ismail, and C. Boyd, ''A survey of trust and reputation systems for online service provision,'' *Decis. Support Syst.*, vol. 43, no. 2, pp. 618–644, Mar. 2007.
[23] S. Ruohomaa, L. Kutvonen, and E. Koutrouli, ''Reputation management survey,'' in *Proc. 2nd Int. Conf. Availability, Rel. Secur. (ARES)*, 2007, pp. 103–111.
[24] H. Yu, Z. Shen, C. Leung, C. Miao, and V. R. Lesser, ''A survey of multi-agent trust management systems,'' *IEEE Access*, vol. 1, pp. 35–50, 2013.
[25] M. John and E. Govindaraj, ''A survey on trust management in peer to peer systems,'' *Int. J. Comput. Technol.*, vol. 1, no. 2, 2014.
[26] S. Srivastava and K. Johari, ''A survey on reputation and trust management in wireless sensor network,'' *Int. J. Sci. Res. Eng. Technol.*, vol. 1, pp. 139–149, Aug. 2012.

[27] H. Mousa, S. B. Mokhtar, O. Hasan, O. Younes, M. Hadhoud, and L. Brunie, "Trust management and reputation systems in mobile participatory sensing applications: A survey," *Comput. Netw.*, vol. 90, pp. 49–73, Oct. 2015.

[28] L. Mui, A. Halberstadt, and M. Mohtashem, "Notions of reputation in multi-agents systems: A review," in *Proc. 1st Int. Joint Conf. Auto. Agents Multiagent Syst. (AAMAS)*, 2002, pp. 280–287.

[29] F. Almenárez, A. Marín, C. Campo, and C. Garcia, "PTM: A pervasive trust management model for dynamic open environments," in *Proc. 1st Workshop Pervas. Secur., Privacy Trust*, vol. 5, 2004, pp. 1–8.

[30] T. D. Huynh, N. R. Jennings, and N. R. Shadbolt, "An integrated trust and reputation model for open multi-agent systems," *Auton Agent Multi-Agent Syst*, vol. 13, no. 2, pp. 119–154, Sep. 2006.

[31] F. G. Marmol, G. M. Perez, and A. F. G. Skarmeta, "TACS, a trust model for P2P networks," *Wireless Pers. Commun.*, vol. 51, no. 1, pp. 153–164, Oct. 2009.

[32] E. Anceaume, G. Guette, P. Lajoie-Mazenc, N. Prigent, and V. V. T. Tong, "A privacy preserving distributed reputation mechanism," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2013, pp. 1951–1956.

[33] P. Lajoie-Mazenc, E. Anceaume, G. Guette, T. Sirvent, and V. V. T. Tong, "Efficient distributed privacy-preserving reputation mechanism handling non-monotonic ratings," CNRS, Paris, France, Tech. Rep., 2015.

[34] M. F. Kaashoek and D. R. Karger, "Koorde: A simple degree-optimal distributed hash table," in *Peer-to-Peer Systems II*. Berlin, Germany: Springer, 2003, pp. 98–107.

[35] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for Internet applications," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 4, pp. 149–160, 2001.

[36] E. Pavlov, J. Rosenschein, and Z. Topol, "Supporting privacy in decentralized additive reputation systems," in *Trust Management* (Lecture Notes in Computer Science), vol. 2995, S. Poslad, C. Jensen, and T. Dimitrakos, Eds. Berlin, Germany: Springer-Verlag, 2004, pp. 108–119.

[37] S. Dolev, N. Gilboa, and M. Kopeetsky, "Efficient private multi-party computations of trust in the presence of curious and malicious users," *J. Trust Manage.*, vol. 1, no. 1, p. 8, 2014.

[38] O. Hasan, L. Brunie, and E. Bertino, "Preserving privacy of feedback providers in decentralized reputation systems," *Comput. Secur.*, vol. 31, no. 7, pp. 816–826, Oct. 2012.

[39] T. Dimitriou and A. Michalas, "Multi-party trust computation in decentralized environments in the presence of malicious adversaries," *Ad Hoc Netw.*, vol. 15, pp. 53–66, Apr. 2014.

[40] L. Mekouar, Y. Iraqi, and R. Boutaba, "Peer-to-peer's most wanted: Malicious peers," *Comput. Netw.*, vol. 50, no. 4, pp. 545–562, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128605002197

[41] D. Fraga, Z. Bankovic, and J. Moya, "A taxonomy of trust and reputation system attacks," in *Proc. IEEE 11th Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Jun. 2012, pp. 41–50.

[42] C. Dellarocas, "Mechanisms for coping with unfair ratings and discriminatory behavior in online reputation reporting systems," in *Proc. 21st Int. Conf. Inf. Syst. (ICIS)*, 2000, pp. 520–525.

[43] A. Jøsang and J. Golbeck, "Challenge for robust trust and reputation systems," in *Proc. 5th Int. Workshop Secur. Trust Manage.* Amsterdam, The Netherlands: Elsevier, 2009, p. 52.

[44] X. Lou and K. Hwang, "Collusive piracy prevention in P2P content delivery networks," *IEEE Trans. Comput.*, vol. 58, no. 7, pp. 970–983, Jul. 2009.

[45] M. Tavakolifard and K. C. Almeroth, "A taxonomy to express open challenges in trust and reputation systems," *J. Commun.*, vol. 7, no. 7, pp. 538–551, 2012.

[46] J. Douceur, "The sybil attack," in *Proc. Revised Papers 1st Int. Workshop Peer-to-Peer Syst. (IPTPS)*, 2002, pp. 251–260.

[47] H. Yu, B. G. Phillip, M. Kaminsky, and F. Xiao, "Sybillimit: A near-optimal social network defense against sybil attacks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2008, pp. 3–17.

[48] K. Hoffman, D. Zage, and C. Nita-Rotaru, "A survey of attack and defense techniques for reputation systems," *ACM Comput. Surv.*, vol. 42, no. 1, pp. 1–31, 2009.

[49] R. Dennis and G. Owen, "Rep on the block: A next generation reputation system based on the blockchain," in *Proc. IEEE 10th Int. Conf. Internet Technol. Secured Trans.*, Dec. 2015, pp. 131–138.

[50] I. Bohnet and R. Zeckhauser, "Trust, risk and betrayal," *J. Econ. Behav. Org.*, vol. 55, no. 4, pp. 467–484, Dec. 2004.

[51] C. Cachin, "Distributing trust on the Internet," in *Proc. Int. Conf. Dependable Syst. Netw.*, 2001, pp. 183–192.

[52] N. El Ioini and C. Pahl, "Trustworthy orchestration of container based edge computing using permissioned blockchain," in *Proc. Int. Conf. Internet Things, Syst., Manage. Secur.*, 2018.

[53] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani, "Applications of blockchains in the Internet of Things: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1676–1717, 2nd Quart., 2019.

[54] A. Bellini, E. Bellini, M. Gherardelli, and F. Pirri, "Enhancing IoT data dependability through a blockchain mirror model," *Future Internet*, vol. 11, no. 5, p. 117, May 2019.

[55] S. Brotsis, N. Kolokotronis, K. Limniotis, S. Shiaeles, D. Kavallieros, E. Bellini, and C. Pavué, "Blockchain solutions for forensic evidence preservation in IoT environments," in *Proc. IEEE Conf. Netw. Softwarization (NetSoft)*, Jun. 2019, pp. 110–114.

[56] E. Bellini, P. Ceravolo, and E. Damiani, "Blockchain-based e-vote-as-a-service," in *Proc. IEEE 12th Int. Conf. Cloud Comput. (CLOUD)*, Jul. 2019, pp. 484–486.

[57] B. Shala, U. Trick, A. Lehmann, B. Ghita, and S. Shiaeles, "Blockchain-based trust communities for decentralized M2M application services," in *Advances on P2P, Parallel, Grid, Cloud and Internet Computing* (Lecture Notes on Data Engineering and Communications Technologies), vol. 24, F. Xhafa, F. Y. Leu, M. Ficco, and C. T. Yang, Eds. Cham, Switzerland: Springer, 2018.

[58] E. Bellini, "A blockchain based trusted persistent identifier system for big data in science," *Found. Comput. Decis. Sci.*, vol. 44, no. 4, pp. 351–377, Dec. 2019.

[59] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, 1982.

[60] M. J. Fischer, N. A. Lynch, and M. S. Paterson, "Impossibility of distributed consensus with one faulty process," *J. ACM*, vol. 32, no. 2, pp. 374–382, 1985.

[61] M. Walport, "Distributed ledger technology: Beyond blockchain," Government Office Sci., London, U.K., Tech. Rep. GS/16/1, 2015.

[62] S. Deetman. (2020). *Bitcoin Could Consume as Much Electricity as Denmark*. [Online]. Available: http://motherboard.vice.com/read/bitcoin-couldconsume-as-much-electricity-as-denmark-by-2020

[63] D. Ongaro and J. K. Ousterhout, "In search of an understandable consensus algorithm," in *Proc. USENIX Annu. Tech. Conf. (USENIX ATC)*, 2014, pp. 305–319.

[64] J.-P. Martin and L. Alvisi, "Fast byzantine consensus," *IEEE Trans. Dependable Secure Comput.*, vol. 3, no. 3, pp. 202–215, Jul. 2006.

[65] J. Yli Huumo, D. Ko, S. Choi, S. Park, and K. Smolander, "Where is current research on blockchain technology?—A systematic review," *PLoS ONE*, vol. 11, no. 10, 2016, Art. no. e0163477.

[66] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proc. 3rd USENIX Symp. Oper. Syst. Design Implement. (OSDI)*, 1999, pp. 173–186.

[67] L. Lamport, "The part-time parliament," *ACM Trans. Comput. Syst.*, vol. 16, no. 2, pp. 133–169, 1998.

[68] M. Cinque, C. Esposito, and S. Russo, "Trust management in fog/edge computing by means of blockchain technologies," in *Proc. IEEE Conf. Internet Things, Green Comput. Commun., Cyber, Phys. Social Comput., Smart Data, Blockchain, Comput. Inf. Technol., Congr. Cybermatics*, Jul. 2018, pp. 1433–1439.

[69] A. Poelstra. (2014). *Distributed Consensus From Proof of Stake is Impossible*. Accessed: Dec. 2018. [Online]. Available: https://download.wpsoftware.net/bitcoin/old-pos.pdf

[70] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton, NJ, USA: Princeton Univ. Press, 2016.

[71] G. Galas. (May 2018). *Analyse et Comparaison des mécanismes de Consensus dans la blockchain*. [Online]. Available: https://medium.com/@godefroy.galas/analyse-et-comparaison-des-mecanismes-de-consensus-dans-la-blockchain-f91aee511ea3

[72] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. Accessed: Dec. 2018. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[73] S. Popov. (2018). *IOTA: The Tangle*. Accessed: Dec. 2018. [Online]. Available: https://iota.org/IOTA Whitepaper.pdf

[74] R. C. Merkle, "Protocols for public key cryptosystems," in *Proc. IEEE Symp. Secur. Privacy*, Apr. 1980, pp. 122–134.
[75] I. Eyal, A. E. Gencer, E. G. Sirer, and R. V. Renesse, "Bitcoin-NG: A scalable blockchain protocol," in *Proc. USENIX Conf. Netw. Syst. Design Implement.*, 2016, pp. 45–59.
[76] N. Szabo. (1994). *Smart Contracts*. [Online]. Available: http://szabo.best.vwh.net/smart.contracts.html
[77] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, vol. 2, no. 9, 1997.
[78] T. Haerder and A. Reuter, "Principles of transaction-oriented database recovery," *ACM Comput. Surv.*, vol. 15, no. 4, pp. 287–317, Dec. 1983.
[79] *Solidity*. Accessed: Dec. 2018. [Online]. Available: https://solidity.readthedocs.io/en/v0.5.3/
[80] M. Herlihy, "Blockchains from a distributed computing perspective," *Commun. ACM*, vol. 62, no. 2, pp. 78–85, Jan. 2019.
[81] U. W. Chohan, "The decentralized autonomous organization and governance issues," *Regulation Financial Inst. eJournal, Social Sci. Res. Netw.*, to be published.
[82] I. Sergey and A. Hobor, "A concurrent perspective on smart contracts," in *Financial Cryptography and Data Security* (Lecture Notes in Computer Science), vol. 10323, M. Brenner, Ed. Cham, Switzerland: Springer, 2017, pp. 478–493.
[83] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on Ethereum smart contracts SoK," in *Proc. 6th Int. Conf. Principles Secur. Trust*, vol. 10204, 2017, pp. 164–186.
[84] K. S. Khan, R. Kunz, J. Kleijnen, and G. Antes, "Five steps to conducting a systematic review," *J. Roy. Soc. Med.*, vol. 96, no. 3, pp. 118–121, Mar. 2003.
[85] R. Wille, "Restructuring lattice theory: An approach based on hierarchies of concepts," in *Formal Concept Analysis*, S. Ferré and S. Rudolph, Eds. Dordrecht, The Netherlands: Reidel, 2009, pp. 445–470.
[86] F. Baader and B. Sertkaya, "Applying formal concept analysis to description logics," in *Proc. ICFCA*, in Lecture Notes in Computer Science, vol. 2961, P. Eklund, Ed. Berlin, Germany: Springer, 2004, pp. 261–286.
[87] J. Poelmans, S. O. Kuznetsov, D. I. Ignatov, and G. Dedene, "Formal concept analysis in knowledge processing: A survey on models and techniques," *Expert Syst. Appl.*, vol. 40, no. 16, pp. 6601–6623, Nov. 2013.
[88] R. C. Nickerson, U. Varshney, and J. Muntermann, "A method for taxonomy development and its application in information systems," *Eur. J. Inf. Syst.*, vol. 22, no. 3, pp. 336–359, May 2013.
[89] L. Mekouar, Y. Iraqi, and R. Boutaba, "Reputation-based trust management in peer-to-peer systems: Taxonomy and anatomy," in *Handbook of Peer-to-Peer Networking*, X. Shen, H. Yu, J. Buford, and M. Akon, Eds. Boston, MA, USA: Springer, 2009.
[90] G. Suryanarayana and R. Taylor, "A survey of trust management and resource discovery technologies in peer-to-peer applications," Inst. Softw. Res., Univ. California, Irvine, CA, USA, Tech. Rep. UCI-ISR-04-6, 2004.
[91] S. Marti and H. Garcia-Molina, "Taxonomy of trust: Categorizing P2P reputation systems," *Comput. Netw.*, vol. 50, no. 4, pp. 472–484, Mar. 2006.
[92] F. Hendrikx, K. Bubendorfer, and R. Chard, "Reputation systems: A survey and taxonomy," *J. Parallel Distrib. Comput.*, vol. 75, pp. 184–197, Jan. 2015.
[93] P. Tasca and C. J. Tessone, "Taxonomy of blockchain technologies. Principles of identification and classification," 2018, *arXiv:1708.04872*. [Online]. Available: https://arxiv.org/abs/1708.04872
[94] H. Wang, Z. Zheng, S. Xie, H. N. Dai, and X. Chen, "Blockchain challenges and opportunities: A survey," *Int. J. Web Grid Services*, vol. 14, no. 4, p. 352, 2018.
[95] E. Kazan, C.-W. Tan, and E. T. K. Lim, "Towards a framework of digital platform disruption: A comparative study of centralized & decentralized digital payment providers," in *Proc. 25th Australas. Conf. Inf. Syst.*, 2014, pp. 1–10. [Online]. Available: https://research.cbs.dk/en/publications/towards-a-framework-of-digital-platform-disruption-a-comparative-
[96] M. Morisse, "Cryptocurrencies and bitcoin: Charting the research landscape," in *Proc. Amer. Conf. Inf. Syst.*, 2015, pp. 1–16.
[97] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. Felten, "Research perspectives and challenges for bitcoin and cryptocurrencies," in *Proc. IEEE Symp. Secur. Privacy*, May 2015, pp. 104–121.
[98] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 2084–2123, 3rd Quart., 2016.
[99] G. Fridgen, F. Regner, A. Schweizer, and U. Nils, "Don't slip on the ICO—A taxonomy for a blockchain-enabled form of crowdfunding," in *Proc. Eur. Conf. Inf. Syst. (ECIS)*, 2018, pp. 1–17.
[100] V. Buterin. (2015). *On Public and Private Blockchains*. [Online]. Available: https://www.coindesk.com/vitalik-buterin-on-public-and-private-blockchains
[101] Y. Alowayed, M. Canini, P. Marcos, M. Chiesa, and M. Barcellos, "Picking a partner: A fair blockchain based scoring protocol for autonomous systems," in *Proc. ACM Appl. Netw. Res. Workshop*, 2018, pp. 33–39.
[102] *POA Network Whitepaper*. Accessed: Dec. 2018. [Online]. Available: https://github.com/poanetwork/wiki/wiki/POA-Network-Whitepape
[103] P4Titan. (May 2014). *Slimcoin a Peer-to-Peer Crypto-Currency With Proof-of-Burn. Mining Without Powerful Hardware*. [Online]. Available: https://goo.gl/EcKpsr
[104] (Feb. 2018). *NEM Technical Reference*. [Online]. Available: https://www.nem.io/wp-content/themes/nem/files/NEM_techRef.pdf
[105] V. Buterin. (Jan. 2014). *Slasher: A Punitive Proof-of-Stake Algorithm*. [Online]. Available: https://blog.ethereum.org/2014/01/15/slasher-a-punitive-proof-of-stake-algorithm/
[106] S. Dziembowski, S. Faust, V. Kolmogorov, and K. Pietrzak, "Proofs of space," in *Proc. Annu. Cryptol. Conf.* Berlin, Germany: Springer-Verlag, 2015. [Online]. Available: https://eprint.iacr.org/2013/796
[107] R. Kotla, A. Clement, E. L. Wong, L. Alvisi, and M. Dahlin, "Zyzzyva: speculative Byzantine fault tolerance," in *Proc. ACM Commun.*, vol. 11, 2008, pp. 86–95.
[108] J.-P. Martin and L. Alvisi, "Fast Byzantine consensus," *IEEE Trans. Dependable Secure Comput.*, vol. 3, no. 3, pp. 202–215, Jul. 2006.
[109] S. Liu, C. Cachin, V. Quéma, and M. Vukolic, "XFT: Practical fault tolerance beyond crashes," in *Proc. 12th USENIX Symp. Oper. Syst. Design Implement.*, 2015, pp. 485–500.
[110] V. M. and D. Y. Miyamoto Wakahara, "Reputation scoring system using an economic trust model: A distributed approach to evaluate trusted third parties on the Internet," in *Proc. 27th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, 2013, pp. 730–737.
[111] D. Carboni, "Feedback based reputation on top of the bitcoin blockchain," 2015, *arXiv:1502.01504*. [Online]. Available: https://arxiv.org/abs/1502.01504
[112] J. Gattermayer and P. Tvrdik, "Blockchain-based multi-level scoring system for P2P clusters," in *Proc. 46th IEEE Int. Conf. Parallel Process. Workshops (ICPPW)*, Aug. 2017, pp. 301–308.
[113] B. Matthew, E. Manosai, H. Christopher, and W. Defu, "Decentralized reputation system for transaction networks," Dept. CIS-Senior Des., Univ. Pennsylvania, Philadelphia, PA, USA, Tech. Rep., 2015. Accessed: Dec. 2018. [Online]. Available: https://www.seas.upenn.edu/~cse400/CSE400_2014_2015/reports/07_report.pdf
[114] A. Schaub, R. Bazin, O. Hasan, and L. Brunie, "A trustless privacy-preserving reputation system," in *Proc. IFIP Int. Inf. Secur. Privacy Conf. (IFIT SEC)*, 2016, pp. 398–411.
[115] M. Sharples and J. Domingue, "The blockchain and kudos: A distributed system for educational record, reputation and reward," in *Adaptive and Adaptable Learning* (Lecture Notes in Computer Science), vol. 9891, K. Verbert, M. Sharples, and T. Klobučar, Eds. Cham, Switzerland: Springer, 2016.
[116] S. Delgado-Segura, C. Tanas, and J. Herrera-Joancomartí, "Reputation and reward: Two sides of the same bitcoin," *Sensors*, vol. 16, no. 6, p. 776, May 2016.
[117] J. Zou, B. Ye, L. Qu, Y. Wang, M. A. Orgun, and L. Li, "A proof-of-trust consensus protocol for enhancing accountability in crowdsourcing services," *IEEE Trans. Services Comput.*, vol. 12, no. 3, pp. 429–445, May 2019.
[118] M. Li, J. Weng, A. Yang, W. Lu, Y. Zhang, L. Hou, J.-N. Liu, Y. Xiang, and R. H. Deng, "CrowdBC: A blockchain-based decentralized framework for crowdsourcing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 6, pp. 1251–1266, Jun. 2019.
[119] M. Axel, D. Benoit, and B. Jean-Luc, "Blockchain based trust & authentication for decentralized sensor networks," 2017, *arXiv: 1706.01730*. [Online]. Available: https://arxiv.org/abs/1706.01730

[120] R. D. Pietro, X. Salleras, M. Signorini, and E. Waisbard, "A blockchain-based trust system for the Internet of Things," in *Proc. ACM Symp. Access Control Models Technol. (SACMAT)*, 2018, pp. 77–83.
[121] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1495–1505, Apr. 2019.
[122] Z. Lu, Q. Wang, G. Qu, and Z. Liu, "BARS: A blockchain-based anonymous reputation system for trust management in VANETs," in *Proc. IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun., 12th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2018, pp. 98–103.
[123] V. Strobel and M. Dorigo, "Blockchain technology for robot swarms: A shared knowledge and reputation management system for collective estimation," Institut de Recherches Interdisciplinaires et de Developpements en Intelligence Articielle, Univ. Libre de Bruxelles, Brussels, Belgium, Tech. Rep. TR/IRIDIA/2018-009, May 2018.
[124] D. Calvaresi, V. Mattioli, A. Dubovitskaya, A. F. Dragoni, and M. I. Schumacher, "Reputation management in multi-agent systems using permissioned blockchain technology," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. (WI)*, Dec. 2018, pp. 719–725.
[125] N. Alexopoulos, S. M. Habib, and M. Mühlhäuser, "Towards secure distributed trust management on a global scale: An analytical approach for applying distributed ledgers for authorization in the IoT," in *Proc. ACM Workshop IoT Secur. Privacy*, 2018, pp. 49–54.
[126] *Namecoin*. Accessed: Dec. 2018. [Online]. Available: https://namecoin.org/
[127] A. Jøsang, S. Hird, and E. Faccer, "Simulating the effect of reputation systems on e-markets," in *Proc. 1st Int. Conf. Trust Manage. (iTrust)*, P. Nixon and S. Terzis, Eds. Berlin, Germany: Springer-Verlag, 2003, pp. 179–194.
[128] M. Kacer, D. Langr, and P. Tvrdík, "Clondike: Linux cluster of non-dedicated workstations," in *Proc. IEEE Int. Symp. Cluster Comput. Grid (CCGrid)*, vol. 1, May 2005, pp. 574–581.
[129] M. Štava and P. Tvrdík, "Overlapping non-dedicated clusters architecture," in *Proc. IEEE Int. Conf. Comput. Eng. Technol.*, vol. 1, Jan. 2009, pp. 3–10.
[130] *Monetha—White Paper*. Accessed: Dec. 2018. [Online]. Available: https://www.monetha.io/about
[131] (2017). *REPU Smart Reputation Management*. Accessed: Dec. 2018. [Online]. Available: https://repu.io/
[132] *UTEMIS*. Accessed: Dec. 2018. [Online]. Available: https://www.utemis.com
[133] *Drep*. Accessed: Dec. 2018. [Online]. Available: https://www.drep.org/
[134] *Dream*. Accessed: Dec. 2018. [Online]. Available: https://blog.dream.ac/blockchain-identity-reputation-systems-2/
[135] *Atonomi*. Accessed: Dec. 2018. [Online]. Available: https://atonomi.io
[136] *Enigma*. Accessed: Dec. 2018. [Online]. Available: https://enigma.co/
[137] C. Chad. *How to Build a Reputation System on Blockchain? Bitconch White Paper Gives Out an Great Answer*. Accessed: Dec. 2018. [Online]. Available: https://medium.com/coinmonks/how-to-build-a-reputation-system-on-blockchain-957bd9ec1ab2
[138] T. McConaghy. (2018). *Bigchaindb 2.0 the Blockchain Database*. Accessed: Dec. 2018. [Online]. Available: https://www.bigchaindb.com/whitepaper/
[139] A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill, "Order-preserving symmetric encryption," in *Proc. EUROCRYPT*, 2009, pp. 224–241.
[140] W. Armstrong, "Dependency structures of database relationships," in *Proc. IFIP Congr.*, 1974, pp. 580–583.
[141] C. Gorenflo, S. Lee, L. Golab, and S. Keshav, "Fastfabric: Scaling hyperledger fabric to 20,000 transactions per second," in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency*, May 2019, pp. 455–463.

EMANUELE BELLINI (Member, IEEE) received the M.Sc. degree *(cum laude)* in communication science from the University of Siena, Italy, in 2006, and the Ph.D. degree in telematics and information society from the University of Florence, Italy, in 2012. He is currently a Research Fellow with the Centre of Cyber Physical Systems, Khalifa University, UAE, and an Assistant Professor with the University of Campania "Vanvitelli," Italy. His research interests include distributed ledger technologies, distributed systems, security and resilience of cyber-physical systems, formal methods, HMI, and knowledge management. He is a member of the IEEE Computer Society—Special Technical Community on Blockchain, IEEE Systems Council, EU Blockchain Observatory and Forum, and European Safety and Reliability Association (ESRA)—TC Human Reliability and TC Risk Assessment. He served as a Deputy Coordinator of H2020-RESOLUTE project on resilience of urban transport system and participates as Task, WP Leader in several national and international research projects. He is also a Deputy of the European Integrated Mission Group for Security (IMG-S) WG on Resilience of Cyber Physical Systems, the President of LOGOS Research and Innovation Association, and the Chair of IEEE CSR Conference.

YOUSSEF IRAQI (Senior Member, IEEE) is currently an Associate Professor with the ECE Department, Khalifa University, UAE. He was the Chair of the Computer Science Department, Dhofar University, Oman, for four years. From 2004 to 2005, he was a Research Assistant Professor with the David R. Cheriton School of Computer Science, University of Waterloo, Canada. He has published more than 110 research articles in international journals and refereed conference proceedings. His research interests include adaptive resource management in multimedia wireless networks, trust and reputation management, cloud computing, and stylometry. In 2008, he received the IEEE Communications Society Fred W. Ellersick Paper Award in the Field of communications systems. He is on many technical program committees of international conferences and always approached for his expertise by international journals in his field.

ERNESTO DAMIANI (Senior Member, IEEE) is the Senior Director of the Artificial Intelligence and Intelligent Systems Institute, Khalifa University, a Leader of the big data area at Etisalat British Telecom Innovation Center, the President of the Consortium of Italian Computer Science Universities (CINI), and Full Professor with UniversitÀ degli Studi di Milano, where he leads the SESAR Lab. His work has more than 16 100 citations on Google Scholar and more than 6 300 citations on Scopus, with an H-index of 34. His areas of interest include artificial intelligence, machine learning, big data analytics, edge/cloud security and performance, and cyber-physical systems. He was a recipient of the Stephen Yau Award from the Service Society, the Outstanding contributions Award from IFIP TC2, the Chester-Sall Award from IEEE IES, and a doctorate honoris causa from INSA Lyon, France, for his contribution to big data teaching and research.