Integrating Blockchain and Internet of Things Systems: A Systematic Review on Objectives and Designs

Nguyen Khoi Tran[a,b,*], M. Ali Babar[a,b], Jonathan Boan[c]

[a] *School of Computer Science, The University of Adelaide, Australia*
[b] *Centre for Research on Engineering Software Technology (CREST)*
[c] *Defence Science and Technology Group, Australia*

Abstract

Recent years have witnessed the emergence of the Internet of Things (IoT) systems that incorporate blockchain (BC) elements in their architecture. Due to discrepancies between the requirements of IoT systems and the characteristics of BC networks, the motivations and design of these blockchain-enabled IoT systems (BC-IoT) are not only intriguing from a research perspective but also invaluable in practice. This paper presents an inductive study of the "why" and "how" of BC-IoT systems through a Systematic Literature Review of 120 peer-reviewed studies. To capture the diverse nature of BC-IoT integration, we proposed and applied a multi-perspective framework to analyse the existing systems. Regarding their motivations, we studied the improvement objectives and technical problems that drive the integration of BC. Regarding the design, we captured the position of BC within IoT systems as well as the content and processes that IoT systems offload to BC. As these dimensions are not mutually exclusive, they constitute a rich and multi-angle view of BC-IoT integration. Based on these findings, we defined 10 archetypes of BC-IoT systems that embody the core patterns of usage and configuration of BC in IoT systems.

Keywords: Blockchain, Distributed Ledger, Smart Contract, Web of Things, Internet of Things, Systematic Review
2020 MSC: 00-01, 99-00

1. Introduction

At the beginning of 2009, Satoshi Nakamoto mined the genesis block of Bitcoin[1] the number of Internet-connected devices surpassed the World's population for the first time, according to an estimation by Cisco [1]. These events mark the birth of two technologies that were hailed as "disruptive" by technologist, enterprises, and legislators alike – *Blockchain (BC)* technology and the *Internet of Things (IoT)*. The IoT technologies empower physical entities – "things" – to observe their environments and themselves, and share this information with computer systems via the Internet. The BC technologies allow mutually distrusting parties, such as IoT-enabled things, to cooperate and exchange value in a verifiable manner without relying on intermediaries [2]. Thus, the convergence of the two technologies was imminent. In the industry, a recent survey of over 500 IoT adopters in the U.S. by Gartner [3] reveals that over 75% of the companies have already adopted or are planning to adopt BC by the end of 2020. In academia, over 800 peer-reviewed research articles about the integration between IoT and BC have been published by 2020. These statistics suggest the emergence of *IoT systems that integrate BC elements in their architecture*. In this paper, we denote these systems as Blockchain-enabled Internet of Things systems (BC-IoT).

Why do IoT systems integrate BC, and how such an integration has been carried out? On the surface, the discrepancy between IoT systems and BC appears irreconcilable: the former generate a large amount of data quickly while latter tend to have severely limited throughputs; the former comprise resource-constrained devices while the latter tend to be resource-intensive. Therefore, the point of BC-IoT research is to justify the integration and develop architectures and mechanisms for it. These insights on "why" and "how" of BC-IoT systems are valuable because they can form a "playbook" to guide the integration of BC that happens in other domains.

Numerous reviews and position papers on the why and how of BC-IoT systems exist [4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]. One of the approaches is to align use cases of BC beyond cryptocurrencies with IoT-related scenarios such as "smart property", "smart home", "smart cities" and deduce the motivations of BC-IoT integration. Another approach to justify the integration is aligning well-known requirements of IoT systems, such as privacy and security, with the potential benefits of BC systems. For instance, Makhdoom et al., [13] proposed a taxonomy of security and performance requirements of IoT systems, and used it

*Corresponding author
Email addresses: nguyen.tran@adelaide.edu.au (Nguyen Khoi Tran), ali.babar@adelaide.edu.au (M. Ali Babar)

[1] https://www.blockchain.com/btc/block/000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f

to assess the impacts of BC technologies on IoT and evaluate four BC platforms (Bitcoin, Ethereum, Fabric, and IOTA). Sengupta et al., [14] reviewed and classified attacks on IoT systems based on objects of vulnerability (i.e., devices, network, software, and data). This taxonomy then served as the security requirements of IoT systems to study the prospects of BC integration. The common feature of these papers is that they approach the topic deductively, meaning that they infer the purposes and mechanisms of BC-IoT systems based on first principles, such as oft-cited requirements of IoT systems and benefits of BC.

In this paper, we inductively study BC-IoT systems, with a focus on their design. Based on the concrete decisions taken by the existing BC-IoT systems in the academic literature, we infer the motivations and the designs of BC-IoT integration. Thus, the results reflect the actuality of the domain and offer a complementary perspective to the potentiality captured by the inductive work. To this end, we applied the Systematic Literature Review (SLR) method to extract and synthesise information from the BC-IoT systems reported in the peer-reviewed literature. A multi-perspective framework was developed to capture the motivations and design of these systems. Regarding the motives (i.e., "why"), we capture what researchers aim to improve or add to IoT systems with BC (i.e., objective) and the technical pain points that researchers seek to solve with BC (i.e., technical problem). Regarding the design of BC-IoT systems (i.e., "how"), we study their architecture in terms of where BC fits in IoT systems, what IoT systems offload to BC, what configurations of the integrated BC networks are, and how the integrated BC networks were optimized for IoT. Using this framework, we analysed **120 prominent papers**, which were chosen from *778 related research works* from the first BC-IoT paper in 2015 to January 2020. Based on the analysis of this corpus, we identified **10 archetypes** of BC-IoT systems that embody the patterns of usage and configuration of BC that underlie most existing BC-IoT systems in the literature. These archetypes are useful not only for understanding and classifying the existing BC-IoT solutions but also for guiding the development of new ones.

Our contributions are as follows:

- Proposing a multi-perspective framework for studying BC-IoT systems by breaking them down to different angles
- Identifying and classifying the "why" of BC-IoT systems on two angles – improvement objective and technical problem
- Characterising the "how" of BC-IoT integration from three perspectives – architecture, content, and configuration of BC
- Identifying and classifying the optimizations necessary to fit BC into constraints of IoT systems
- Defining 10 archetypes of BC-IoT integration
- Presenting arguments for and against BC-IoT integration and discussing the short-term and long-term future research of BC-IoT systems

The remainder of this paper is organized as follows. Section 2 provides an overview of IoT, BC, and describe the related work. Section 3 provides the details of our multi-perspective framework. Section 4 presents the protocol that we have used to select and analyze the studies. Section 5 provides answers to the "why" of BC-IoT systems. Section 6 answers the "how" perspective of BC-IoT systems in terms of architecture, content, and configuration of the integrated BC. Section 7 presents and classifies techniques to optimize BC for integration with IoT systems. Finally, we discuss the results and outline some of the areas of the future research in Section 10.

2. Background and Related Work

A Blockchain-integrated IoT system (BC-IoT system) is an IoT system that include blockchain elements in its architecture. Therefore, an understanding of IoT systems' architecture as well as BC networks' structures and operations are necessary for analysing BC-IoT systems. In this section, we elaborate on the background information. Some existing systematic reviews on the topic of BC-IoT systems were also introduced and compared with the current work.

2.1. IoT Systems

For well-known and well-funded concepts such as IoT systems, their definitions are surprisingly varied and elusive. There are at least three angles to define an IoT system, each having a different emphasis [15]. The thing-oriented vision focuses on the "smartening" of physical objects by the mean of embedded sensors, computation, and communication. The Internet-oriented vision focuses on bringing physical devices to TCP/IP networks and addressing the implications of such an influx. The semantic-oriented vision concerns with the addressing, representation, and exchange of information generated by physical devices. For practical purposes, we consider IoT systems as computer systems that utilize electronic tags, sensors, and actuators over the Internet. Our review considers both individual IoT systems as well as system-of-systems that govern cross-orgainzational use cases such as smart cities and supply chain management.

IoT systems generally assume a three-tier Edge-Fog-Cloud architecture. Located at the edge are *sensors* and *actuators* that transform physical stimuli into digital signals and vice versa. These devices are generally connected to microcontroller units (MCU) via Universal Asynchronous Receiver / Transmitter (UART), Serial Peripheral Interface (SPI), or Inter-integrated Circuit (I2C) protocols to form *edge nodes*. These MCU control sensors and actuators, and communicate their data to other components of IoT systems. Based on computing capability, MCU can

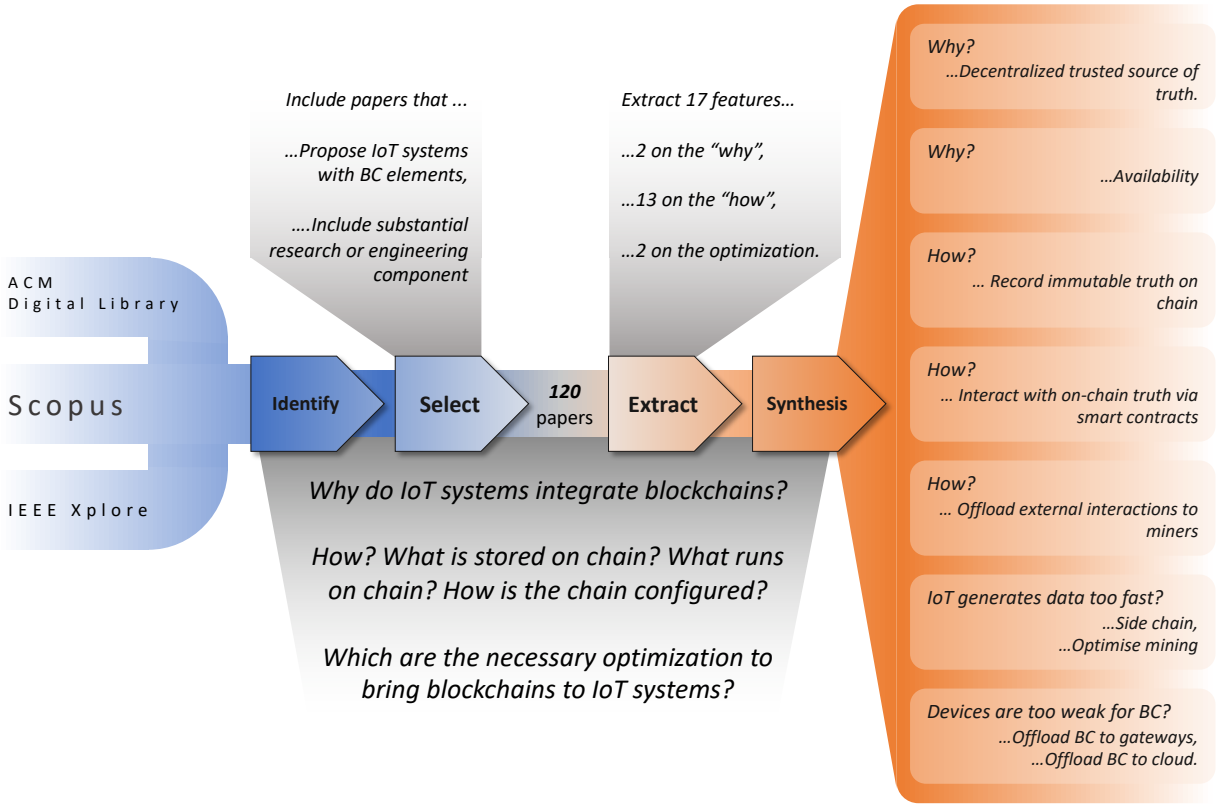

Figure 1: The research questions, review process and key findings.

be organized into three classes [16]. The simplest ones are 8-bit, which are commonly found in most Arduino boards. They tend to operate at a low speed, have less internal Random Access Memory (RAM), support fewer peripherals, and generally do not support operating systems (OS). At the other end of the spectrum are 32- or 64-bit platforms such as Raspberry Pi 4 that support full Linux operating systems. Being capable computers, these platforms can operate BC elements. In this paper, we use the term *Edge Deployment* to denote the deployment of BC elements on these micro-controllers.

Most IoT systems revolve around centralized IoT platforms that reside on clouds or data centers. Each platform provide a unified interface to a diverse and distributed set of edge nodes. This centralization simplifies both the interaction with and the management of IoT devices. For instance, when users issues instructions to their smart home devices via a mobile or a web interface, these commands generally are not routed to the devices. Instead, they modify the states of the digital representations of the devices, hosted on their manufacturers' clouds. Physical devices then synchronize with their digital representations via the Internet. Many BC-IoT systems employ BC elements on the cloud infrastructure or treat BC networks as a remote cloud services. In this paper, we use the term *Cloud Deployment* to denote the deployment of BC elements on cloud-based infrastructure or data centers.

While clouds enable the functionalities of IoT systems, they also limit the response rate of the these systems due to their distance from the edge nodes. Therefore, there is a need for introducing additional computing and storage capacity on-premise at the network's edge. This model of edge processing is defined as fog computing. Nodes that enable fog computing are denoted as *fog nodes*. They communicate with edge nodes via wired or wireless local area network (LAN), as well as wireless personal area network (WPAN) protocols such as ZigBee and bluetooth low energy (BLE). They connect to clouds or data centers via wide area network technologies such as ADSL, WIMAX, 4G, or optical fiber. In this paper, we use the term *Fog Deployment* to denote the deployment of BC elements on fog nodes. Figure 2 depicts a three-tier IoT system with three deployment location highlighted.

2.2. Blockchain

BC has a long academic pedigree, dating back to seminal work on linked timestamping in 1980s [17]. Thus, it is difficult and inadequate to capture BC's essence with a definition. Moreover, there is a big difference between knowing the name of something and knowing something. Therefore, in this section, we present BC from three perspectives: its common definitions, how it works, and its architecture.

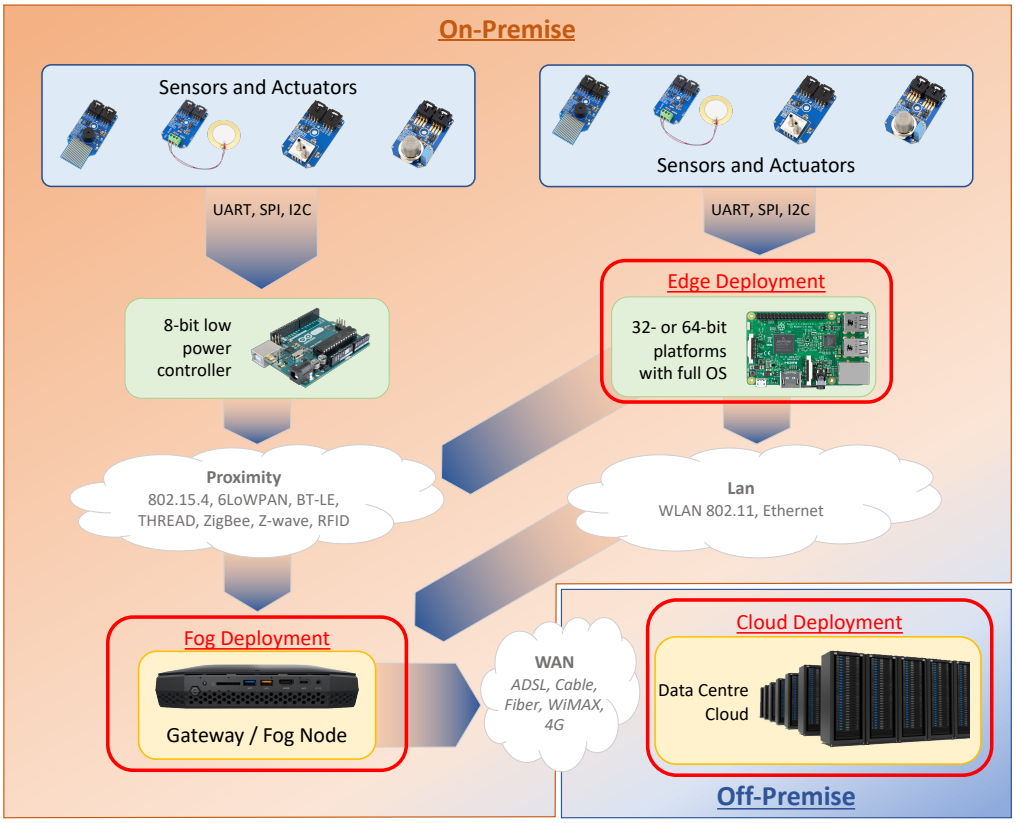

Figure 2: Edge-Fog-Cloud Architecture of IoT Systems.

2.2.1. What is blockchain?

Despite the growing interest and investment in BC, the definition of this technology is still elusive. While BC is commonly known as the technology behind Bitcoin, this terminology did not appear in the white paper of the cryptocurrency system [18]. In fact, Narayanan et al., [17] stated that BC is merely an umbrella term for a class of systems that share some characteristics of Bitcoin. One of such systems is Hyperledger Fabric – an open source BC platform that targets enterprise use cases. It defines BC as an immutable ledger for storing transaction, which is maintained by a network of mutually distrusting peers [19]. This definition echoes the ones appearing in the BC-IoT studies that we have reviewed as well as in the white papers on other BC protocols on the market. From a software engineering perspective, BC can be considered a new type of software connector that enable a decentralized shared storage among distrusting entities and software systems [20].

The common threads among these definitions are the main properties of BC: decentralization, trustless, and immutability. As a decentralized system, BC operates on a network of peers and does not rely on a centralized arbitrator. As a trustless system, the participants in BC rely on asymmetric encryption, digital signature and other cryptography mechanisms to verify incoming transactions instead of trusting others blindly. This network of distrusting peers co-maintains a data structure that has been designed to make tampering detectable, making this data structure (i.e., the ledger) immutable and irreversible. The reviewed BC-IoT systems leveraged these characteristics to bring benefits to or resolve problems of IoT systems.

2.2.2. How Blockchain works

From a technical point of view, *blockchain is a cryptographically secured transactional singleton machine with shared state [21].* As a transactional state machine, a BC network transits between states by processing transactions. Some BC platforms such as Bitcoin uses restricted scripting languages to process transactions. Others employ Turing-complete languages, allowing their users to specify arbitrary transaction processing logic.

As a cryptographically secured system, BCs rely on cryptography for security. Most BC platforms utilize digital signatures to ensure authenticity, integrity, and non-repudiation of the transactions. The collection of all transactions held by a BC is denoted as a *ledger*. To protect the integrity and order of transactions within a ledger, most BC platforms arrange transactions into blocks, each of which contains the cryptographic hash of the previous one. This data structure makes tampering of the historical data visible, as changes would be propagated from the tampering point to the latest ledger block. Therefore, the use of cryptography allows BC participants to verify the

data and the operation of the network instead of merely trusting the party that stores and processes the data.

As a singleton machine with shared state, BC ensures that instances of the state machine held by its participants are identical and at the same state. In other words, a BC network ensures that all participants observe a singular truth. The format and content of the shared truth vary by the design of BC and the use case. BC networks employ consensus protocols to maintain the consistency of the state information among their participants. These protocols dictate the conditions under which a block or a transaction can be considered valid. They also determine the which participant can append transactions to the ledger via a process called "mining." Nakamoto consensus protocol, also called Proof-of-Work (PoW), is one of the most common consensus protocol in BC networks. According to this protocol, any participant who wants to process transactions and update the ledger must solve a cryptographic puzzle to prove that it has expensed effort on the process. This puzzle is hard to solve but easily verified by other participants. The purpose of PoW is preventing Sybil attacks, in which an adversary creates many fake votes to drive the consensus of network. Other BC consensus protocols such as Proof-of-Stake (PoS) and Practical Byzantine Fault Tolerance (pBFT) were applied to circumvent the resource-intensive puzzle solving.

2.2.3. Architecture of a Blockchain Network

A BC network is a peer-to-peer system that operates a BC. Nodes in a BC network are denoted as *BC nodes*. They can be classified into full and lightweight nodes based on the content that they hold and their role in the network. *Full nodes* serve as the backbone of BC networks. They hold complete copies of the ledger and take part in the mining process. Thus, they offer their operators the highest degree of security and autonomy at the price of immense resource consumption. For instance, a full node of a Bitcoin network requires more than 200GB of high-speed storage. *Lightweight nodes* were developed to circumvent this problem. They store only ledgers' headers instead of the entire transaction log. As a result, they cannot verify incoming transactions themselves but must rely on some trusted full nodes. Furthermore, they cannot participate in the mining process. Many BC-IoT systems employ both full and lightweight nodes in their operation. Decisions on the type of BC nodes to use and where to deploy them within an IoT system are some key features to describe a BC-IoT system.

2.3. Related Work

As BC-IoT systems garnered more attention, the interest in positioning and reviewing them has also grown. Using the same systematic selection process that we employed in this review, we have identified related reviews and position papers. A majority of them are narrative-based surveys. Some focused on the motivation of BC-IoT integration and approached the topic deductively. Their premises are benefits brought about by BC [4, 8] and deficits or security threats of IoT systems [5, 14, 10, 13]. By matching the benefits of BC with the needs of IoT, conclusions regarding the motivation of BC-IoT systems can be drawn. Our review offers a more empirical perspective to this topic based on the objectives and problems targeted by concrete BC-IoT systems. The results, thus, reflect the actuality of the domain.

Other narrative-based surveys focused on enumerating and comparing the existing BC platforms on their internal structure [6] and how they satisfy the requirements of IoT systems, such as performance and security [13]. Their results inform developers to choose the most performant and secure BC platform for their BC-IoT systems. However, the selection of a BC platform is only one of the design decisions that developers have to make when building a BC-IoT system. Our review complements these results with insights on other aspects such as parts of IoT system to replace, content to place on BC, and configurations of the integrated BC network.

Beside narrative-based survey, we have identified three systematic review related to BC-IoT systems. Conoscenti et al., [7] focused on understanding whether BC can be employed to enable a "decentralized and private-by-design IoT". Based on 35 papers, this review identified 16 use cases of BC, four of which are specific to IoT. The review also found 17 types of data to be stored on BC, four mining techniques and presented some techniques for integrity preservation, anonymization, and scaling on BC. Panarello et al. [9] proposed to organize BC-IoT papers into four groups including Smart City, Smart Home, Smart Property, and Generic. The review proposed two use patterns of BC in IoT contexts and assessed the level of development of some existing BC applications on a five-level scale. Lo et al., [12] reviewed 35 BC-IoT systems in terms of motivation and design. The review also assessed the evaluation method and metrics used by the existing BC-IoT papers. These assessment features are unique among the identified surveys.

Table 1 compares our review with the existing systematic reviews. Regarding the scope and research questions, our review targets not only the motivation but also the design aspects of BC-IoT systems. While some research questions appear to be similar to the ones from the existing surveys, we studied and addressed them as a greater degree of granularity. For example, we considered the question "what role can blockchain play to address the existing issues of IoT?" from two perspectives – improvement objectives and technical problems. Such information contribute to a richer and more comprehensive picture of BC-IoT systems. Moreover, it allows us to identify and define the archetypes that underlie most BC-IoT systems.

Our review also exceeds the existing reviews in terms of the sample size and the number of studied features. The larger sample helped unveiling not only alternative motivations and solutions of BC-IoT integration but also

Table 1: Comparison between this review and previous systematic surveys on BC-IoT systems.

Review	Size	Features	Primary Question
Conoscenti, et al., 2017 [7]	35 studies	6	Can the blockchain foster a private-by-design IoT?
Panarello, et al., 2018 [9]	51 studies	6	What are usage of BC-related approaches and technologies in IoT context?
Lo, et al., 2019 [12]	35 studies	14	What role can blockchain play to address the existing issues of IoT?
This review	**120 studies**	**17**	*Objectives and Designs of BC-IoT systems*

their relative weights. The larger sample is also conducive to revealing outliers, which represents unique issues and solutions. By assessing more features, our review also unveils more details of BC-IoT systems. For instance, Lo et al. [12] assessed the integrated BC networks on two features, namely the utilized BC platform and the consensus protocol. Our review extended this feature set to capture additional details such as the number of BC networks being employed, the data structure of the ledger maintained by these networks, and the type of global state that these ledgers contain.

Another advantage of a larger feature set is that it allows BC-IoT systems to be analysed from multiple angles. To illustrate this point, let's consider the role of the integrated BC network as an example. Lo et al. [12] studied this topic with one feature – "BC role." and generated noun-phrases such as "data storage" as the results. Our review, on the other hand, assessed the role of integrated BC networks from three different perspectives with eight features. From the perspective of IoT system, we studied the improvement objectives and technical problems that BC is expected to address. Another perspective is where BC fits into an IoT system. We considered both the functional modules of IoT systems that BC networks replace and how they were deployed. The final perspective is on what BC networks store and process on behalf of an IoT system. We employed four features to investigate this perspective: on-chain data and logic, and off-chain data and logic. Together, the large feature set provides a multi-angle view of BC-IoT systems that cover not just motivation but also architectural details to support future research and engineering.

3. Multi-perspective Framework for BC-IoT

BC-IoT systems are complex because they contain design decisions, trade-offs, and technical complexity from both IoT and BC. Therefore, knowledge about these systems is also multi-faceted. Different parties working with BC-IoT systems require different types of knowledge about them.

One aspect of BC-IoT systems is the **"why"** of this BC-IoT integration. This knowledge is relevant to prospective adopters of BC who are considering whether it is the right solution for their existing IoT or the related systems in the domain of cyber-physical and tactical. To be beneficial to these parties, this "why" should be captured in a fine-grained manner, beyond the conventional narrative that BC is "transformative" because it is "immutable" and "transparent". Based on our analysis of the existing BC-IoT studies, there are two perspectives on the "why" of BC-IoT integration. The first one is *objective* of the combination, which captures what researchers aim to improve or add to IoT systems with BC. The second one is technical problems, which capture specific technical pain points of IoT systems that researchers seek to solve with BC. These perspectives are not independent, as solving a technical problem might be the key to achieve an improvement objective and vice versa. Thus, these perspectives complement each other to form a fuller understanding of the "why" of BC-IoT systems.

The other aspect of BC-IoT systems is the **"how"**. This knowledge is beneficial to both researchers and practitioners who aim to bring BC into their existing systems. Because BC-IoT systems combine BC networks with IoT systems, their "how" is complex and requires consideration from multiple perspectives.

For IoT systems, the "how" of BC-IoT integration consists of two perspectives. The first one is about *how BC fits into an IoT system*. This perspective considers both *logical* and *physical* position of BC within an IoT system. The logical location of an integrated BC denotes the functional modules of an IoT system that it replaces or enhances. The physical position indicates BC's nodes reside on the infrastructure of an IoT system. The second perspective is about *what an IoT system offloads to BC*. This perspective captures the type of data and logic that an IoT system deploys on an integrated BC.

For BC, the "how" of BC-IoT integration consists of two perspectives. The first one is about *how integrated BC networks are configured*. This perspective captures critical architectural design decisions of an integrated BC, such as the number of networks being used, the protocol to run, and the technology to develop the integrated BC networks. The second perspective is about *how BC is optimized to fit into IoT systems*. Putting all of these perspectives to-

gether, we formed a multi-perspective framework for BC-IoT systems. Figure 3 visualizes this framework.

In order to show how the framework works, let's consider one of earliest studies on BC-IoT integration that we found in this review as an example. In this study, Zhang and Wen [22] stated that IoT requires an E-business model to incentivise the ownership and exchange of IoT resources, such as sensing data. They pointed out that the involvement of third parties greatly decreases the efficiency and cost-effectiveness of these exchanges. Therefore, they proposed to use blockchain to realize a distributed autonomous corporations (DAC) to orchestrate these business processes. The proposed solution was built based on Bitcoin's protocol. Based on the proposed framework, the descriptions of this BC-IoT system from different perspectives are as the follows:

- *Objective of the integration:* Support the ownership and exchange of resources managed by IoT systems (i.e., renewable energy)
- *Technical problem driving the integration:* Control and incentivize machine-to-machine trading, without relying on a trusted third party.
- *Where BC fits:* logically, BC introduces a business process orchestrator module, which is shared among multiple IoT systems. Physically, the utilized BC network is Bitcoin – an existing public BC network. Thus, from the viewpoint of IoT systems, the BC network is comparable to a remote cloud service.
- *What IoT offload to BC:* The IoT systems store records of resource exchanges on the BC. They also offload the logic of business processes that govern the exchange.
- *BC configuration:* The integrated BC consists of one public BC network, which operates according to the Bitcoin protocol.
- *BC optimization:* No optimization was applied to the integrated BC network.

4. Research Method

In order to systematize the knowledge regarding the motivations and design of BC-IoT systems inductively, we applied the Systematic Literature Review (SLR) research method. An SLR is a secondary study which aims at identifying, evaluating, and synthesising all research relevant to a topic area, a phenomenon of interest, or a research question [23]. In this study, the research questions are the "why" and "how" of BC-IoT systems, the existing BC-IoT prototypes in the literature are pieces of evidence, and the SLR method provides the inductive link between the evidence and the questions. Our review process is organized into four phases according to the guideline by Kitchenham et al., [23]:

1. Identifying potential studies by querying credible sources with a structured query, derived from our research questions
2. Selecting studies based on their quality and relevance to the research questions
3. Extracting data from the chosen studies using features derived from the research questions
4. Synthesizing the data to answer the research questions with the narrative synthesis method

Our SLR was driven by three research questions. They target different aspects of BC-IoT systems described in the multi-perspective framework (Figure 3).

- *(RQ1) Why do IoT systems integrate with BC?* This question aims at the improvement objectives and the technical problems that motivate the existing BC-IoT systems.
- *(RQ2) How do IoT systems integrate BC?* Due to the multi-faceted nature of the BC-IoT integration, we decomposed the RQ2 into three sub-questions: (RQ2.1) How do BC fit with IoT systems? (RQ2.2) What do IoT systems offload to BC? (RQ2.3) What are configurations of the integrated BC?
- *(RQ3) What optimizations were performed on BC to fit them with IoT systems?* This question captures the optimizations done on a BC protocol or architecture for the integration purpose.

4.1. Study Identification

We identified and selected studies for this SLR in a four-step process as depicted in Figure **??**. First, we identified potential studies from bibliographic databases by the means of a structured query. This step concluded in *January, 2020* and resulted in 778 studies. Second, we performed coarse-grained selection based on the titles and abstracts to reduce the number of studies to 226. Third, we conducted fine-grained selection based on the papers' full text to further reduce the number of studies to 159. Finally, we filter studies by the quality and created the final set of 120 studies.

We applied the following cross-validation and adjustment process to minimize inaccuracy and bias in the selection. First, the first author generated and circulated a random sample of papers among co-authors. The co-authors then applied the same selection process without knowing the prior assessment. Afterwards, the authors compared the results and adjusted the set of selected studies if the agreement ratio was less than 85%.

The sources of peer-reviewed literature used by our SLR are Scopus, IEEE Xplore and ACM Digital Library. Scopus is an abstract and citation database. Unlike from Google Scholar, Scopus was not crawled by manually curated by domain experts. By January 2020, it contains over 25,100 titles from more than 5000 publishers world

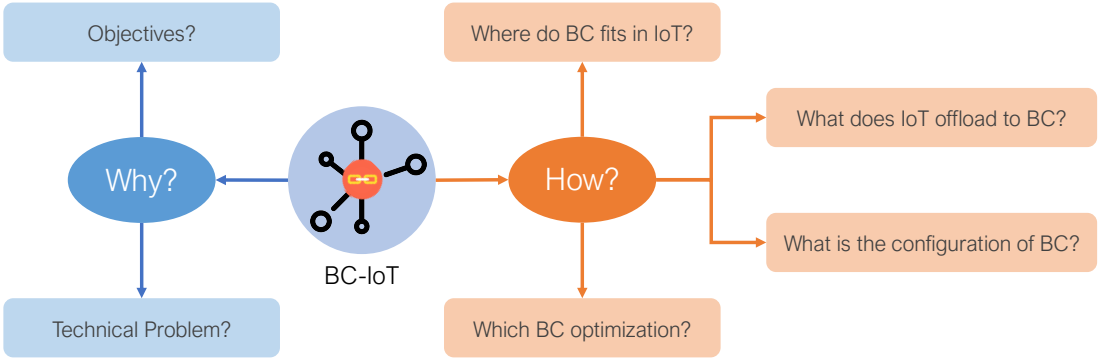

Figure 3: A multi-perspective framework for assessing BC-IoT systems.

wide, including Elsevier, Springer, Wiley-Blackwell, Taylor & Francis, Sage, IEEE, and others[2]. Therefore, with Scopus, we can apply a unified query process across prominent sources of Computer Science literature. This ability increases the comprehensiveness and reliability of the study identification process. We complemented Scopus with IEEE Xplore and ACM Digital Library as these sources contain papers from emerging IoT and BC venues that are yet to be indexed by Scopus.

We utilized the following structured query to retrieve potential studies from the chosen bibliography databases. The query consists of multiple keywords connected by logical operators. Each keyword captures a group of articles that possess matching titles, abstracts, or keywords. Disjunction operations (OR) combine articles matching different keywords into a larger set. Conjunction operations (AND) find the intersection between sets of articles.

> [“Blockchain” OR “block chain”]
> AND
> [“Internet of Things” OR “IoT” OR “Web of Things” OR “WoT”
> OR “Industrial Internet of Things”]

We conducted a series of pilot searches using different combinations of keywords to construct and refine the query. The goal was to identify an analysable set of studies that closely capture the research at the intersection between BC and IoT. The ‘‘blockchain’’ OR ‘‘block chain’’ part of the query identifies research in the domain of BC. These keywords capture two common spellings of BC, which are often mutually exclusive. We decided not to include keywords that denote technical topics underlying BC, such as consensus protocol and distributed networking. These keywords capture research that improves BC platforms generally. While they might mention IoT, they do not represent the concrete BC-IoT systems that this review studies.

We also decided not to include alternative names of BC, such as Distributed Ledger, for two reasons. First, because the term BC has emerged as the canonical name of the concept in recent years, most research would mention it in the abstract or keywords, even if they uses alternative designations. Thus, these research would be picked up by the query. Second, introducing additional keywords to the query inevitably adds noises to the search results. Thus, we must aim for the minimal viable set of keywords for the practical purposes. On the same ground, we rejected keywords such as Cyber-Physical Systems, Ubiquitous Computing, and Embedded Systems. While IoT systems are cyber-physical systems that rely on embedded systems to realise ubiquitous computing, these concepts are not synonyms of IoT.

4.2. Study Selection

The specified query process retrieved potential studies, which can be organized into three types. The first type of papers covers reviews and speculations of how BC and IoT systems work together. The second type of papers present generic improvements to BC, such as new consensus and mining mechanisms, to make BC more compatible with IoT systems. The third type of papers propose specific integration of BC networks into IoT systems to address a concrete purpose, such as improving the access control processes or facilitating machine-to-machine trading. Among these types, on the third one is relevant for our study. Literature reviews are by nature irrelevant for SLR. Generic BC improvement papers are not suitable because they cannot offer concrete evidence regarding the purpose and position of BC networks within IoT systems. Finally, while speculative papers might offer useful insights into the why and how of BC-IoT integration, they lack the “weight” and assurance of papers that present concrete prototypes or simulations.

Based on the stated observations, we formulate the following inclusion (I) and exclusion (E) criteria to select concrete BC-IoT papers. For practical purposes, we con-

[2] https://www.elsevier.com/solutions/scopus/how-scopus-works/content

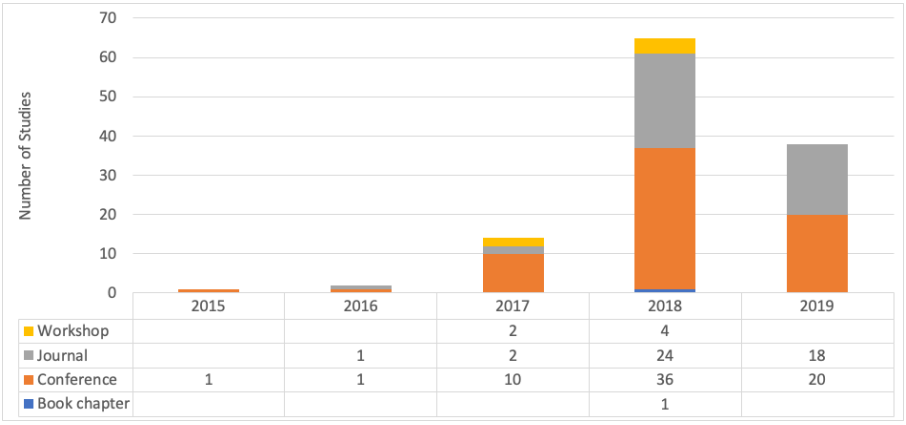

Figure 4: Distribution of the selected studies by publication year and type.

sidered only studies written in English because all authors have an English background.

- **(I1)** Include studies that address specific improvement objectives or technical problems of IoT systems with blockchains and smart contracts
- **(E1)** Exclude studies that present generic improvements to BC, such as new consensus and mining mechanisms
- **(E2)** Exclude primary studies that offer speculations without substantial design or engineering components to back them up
- **(E3)** Exclude secondary studies, short, and position papers
- **(E4)** Exclude patents, books, pre-prints, technical reports, and white papers
- **(E5)** Exclude studies not written in English

Due to the openness and rapid development of the BC domain, we also decided to exclude non-peer-reviewed literature such as patents, books, pre-print, technical reports, and white papers. While certain non-reviewed technical content such as Bitcoin's white paper [18], Ethereum's yellow paper [21] are crucial in the domain of BC, we decided to rely on peer-review as the baseline for the validation of BC-IoT studies. It should be noted that the prestige of publication venues did not contribute to our quality assessment. The reason is that BC-IoT is still a young field, whose first paper appeared in 2015. Thus, BC-IoT studies would be scattered instead of concentrating at high ranking venues. Therefore, we focus on how *concrete* and *substantiated* a study is as a proxy for its quality.

Figure 4 depicts some statistics of the selected studies. The earliest work in the set appeared in 2015. It proposed a business model for exchanging resources in IoT systems using BC as an orchestrator. By 2018, the number of BC-IoT studies grew to 72. On average, 6 out of 10 studies appeared in conferences, and 3 out of 10 studies have been published in journals.

4.3. Extraction Features

We extracted data for *17 features from the 120 selected studies* to answer the research questions. These features were derived primarily from the proposed framework. We extracted the data required for answering the research questions from the reviewed papers and stored the datain a tabular format.

For RQ1, we extracted *improvement objectives* and *technical problems* of IoT systems mentioned in the papers. For RQ2.1, we extracted the *logical and physical position* of BCs within IoT systems. For RQ2.2, we extracted *on-, off-chain data and on-, off-chain logic*. For RQ2.3, we extracted seven features, which characterize the configuration of integrated BC networks. These features include:

- Number of integrated blockchain networks
- Data structure of the ledger
- Type of on-chain global state
- Type of smart contract
- Miner selection protocol
- Blockchain's permission
- Blockchain development technology

Finally, for RQ3, we extracted the *challenges* and the *optimization solutions* for integrating BC into IoT systems. We excluded speculative challenges and reported only the ones from papers that also propose solutions.

5. Why do IoT Systems integrate BC?

The motivation of BC-IoT integration can be expressed in two angles. The first one focuses on gains that BC brings to IoT systems. These improvements can be in

terms of quality attributes or new functionalities. In this review, we denote these improvements as *improvement objectives*, or objective for short. Improving various aspects of security and facilitating exchanges of IoT resources are among the most common objectives of BC-IoT systems. Section 5.1 discusses these objectives in detail.

The second angle of BC-IoT integration focuses on the deficits of IoT systems that researchers employ BC to solve. In this review, we denote these challenges and problems are *technical problems*. The reviewed BC-IoT systems targeted problems regarding the decentralized operation, integrity assurance, incentive, and trustworthy communication channels. Section 5.2 discusses these problems in detail.

5.1. Objectives of Blockchain Integration

Most BC-IoT systems employ BC to modify some existing mechanisms or components of IoT systems to improve their quality attributes. A few leveraged BC and smart contracts to grant IoT systems the ability to do new things. Table 2 and 3 depict quality attributes and functionality that drive BC-IoT integration. Top three objectives of each category are highlighted. Because BC-IoT systems can pursue multiple objectives, the presented frequencies do not add up to 120.

Quality Attribute Improvement: We adopt the software qualities defined in the ISO/IEC 25010 standard to describe quality attributes targeted by the reviewed BC-IoT systems. Security is, by far, the most targeted attribute. For clarity, we decompose the security quality into five components: integrity, authenticity, accountability, confidentiality, and non-repudiation. Driven by the distrust in the ability of cloud services in protecting sensing data and services, driven by their low auditability and numerous data breaches in recent years, integrity improvement was the most common goal of BC-IoT integration. Some BC-IoT systems stored their entire data on BCs [24], while the others kept only the digests of their cloud data on BCs for integrity verification. Some systems leveraged BC networks to store devices' configurations for the future verification [25]. The verification logic can also run directly on BCs as smart contracts, adding another layer of security.

Accountability and non-repudiation are other common attributes targeted by BC-IoT systems, as BC can maintain tamper-proof proofs of responsibility. For instance, Some studies [26, 27, 28] have used BCs to log transactions among and within IoT systems. Other systems extended the hash chaining concept of BCs to create provenance chains of items [29] or security updates for IoT devices [30].

The reviewed BC-IoT systems have also improve authenticity and confidentiality of IoT systems by migrating authentication and access control components from centralized servers to BC and smart contracts, and rely on the mathematics built into BC for security instead of human's assuranceFor instance, Kaga et al., [31] stored biometrics and ran an authentication mechanism directly on a BC network. Wu et al., [32] used a BC network as a second channel for two-factor authentication of IoT devices. A few efforts, reported in [33, 34, 35], have built and deployed authorization mechanisms directly on top of BC networks.

Compatibility, availability, and performance are other attributes targeted by BC-IoT systems. These attributes primarily rely on the peer-to-peer nature of BC networks, which circumvents the single point of failure problem plaguing the centralized models. For instance, Boudguiga et al., [30] and Lee et al., [36] used BCs and smart contracts to build a reliable firmware delivery channel. Duplication among nodes guarantees that the firmware would be available even when devices' manufacturers shut down. Leveraging BC as a trusted communication channel also simplifies the integration of different IoT systems, thus increasing their compatibility.

New Functionality: Around a third of the reviewed BC-IoT systems use BC to add new functionality to IoT systems. These functionalities are primarily based on the ability to manage the ownership and exchange of resources by BC and smart contracts. However, we also found some non-financial use cases of BC and smart contracts, such as event logging and process orchestration.

Orchestrating processes involving IoT devices is the most common functionality sought by the reviewed BC-IoT systems. Both intra- and inter-organization processes were targeted in the existing literature. For instance, BC and smart contracts were used to coordinate the production of renewable energy in a smart grid [37]. This process involves generators, solar panels, and batteries that belong to different organizations. that do not necessarily trust each other. The use of BC allows this process to be carried out without a trusted third party.

The ownership and exchange or resources monitored, generated, or consumed by IoT systems are the most common functionality targeted by BC-IoT systems. Resources monitored by IoT systems include renewable energy and physical assets. Resources generated by IoT systems include sensor data and actuation services offered by the devices. Resources consumed by IoT systems include CPU time, storage space, and software patches. BC and smart contracts provide trustworthy bookkeeping, native cryptocurrencies, and the ability to handle complex business transactions related to these resources. For instance, Kang et al. [38], Li et al. [39], and Pieroni et al. [40] utilized BCs to orchestrate the trading between consumers who can generate electricity. Sun et al. [41] and Lundqvist et al. [42] used BCs to enable devices such as electric cars to purchase electricity. Human-to-machine trading was also considered in ride-sharing schemes, such as [28] and [28].

Maintaining and assessing reputation ratings of devices and services are also common functionality enabled by BC and smart contracts among the reviewed BC-IoT systems. These papers seek to establish secure storage for reputa-

Table 2: Improvement objectives of BC-IoT systems in terms of Quality Attributes

Quality	Description	Frequency
Improve Security in terms of Integrity	Integrity indicates the ability of a system to prevent unauthorised access and modification of data and computer program. Works that use BC as immutable storage of some records or procedures fall into this category.	**74**
Improve Security in terms of Accountability	Accountability denotes the degree which actions or an entity can be traced uniquely to it.	**48**
Improve Security in terms of Non-Repudiation	Non-repudiation is the degree that a system can prevent an actor from denying that an event or action has taken place.	**37**
Improve Security in terms of Confidentiality	Confidentiality is the degree that a system ensures that the data are accessible only to those authorized to have access.	31
Improve Security in terms of Authenticity	Authenticity indicates the degree that a system can verify that the identity of a subject or resource is the one claimed. Works that use BC as authentication mechanisms fall into this category.	27
Improve Compatibility in terms of Interoperability	Interoperability denotes the the degree to which two or more systems, products, or components can exchange information and use the information that has been exchanged. Works that uses BC to bridge IoT systems or devices from different organizations with different formats fall into this category.	7
Reliability in terms of Availability	Availability is the degree that the system is operational and accessible when required for use.	5
Improve Performance in terms of Time Behaviour of the system	Performance denotes the degree to which a system can meet its requirements regarding response processing time and throughput.	5

tion ratings of devices and services of IoT systems (e.g., [43, 44, 45, 46]). They also aim to develop mechanisms for updating these ratings in a transparent and verifiable manner, without relying on centralized service providers (e.g., [47, 48, 49]). BC provides decentralized, tamper-proof source of truth and computation to realize these functionalities.

A few BC-IoT systems utilize these features of BC to realize some less popular functionalities such as service discovery and synchronizing IoT devices within or across IoT systems. For example, Fan et al. [50] used a BC network to provide a trusted source of time for multiple IoT devices. Qiu et al. [51] used a BC network as a source of truth to synchronize the flow control among SDN controllers.

5.2. Problems posed by IoT Systems

The second perspective to the research question RQ1 focuses on the technical problems of IoT systems that the existing BC-IoT systems rely on BC to solve. Table 4 presents these problems in detail along with their frequencies. As BC-IoT systems can address more than one problem, these frequencies do not sum up to 120. For brevity, we organize these problems into five categories: ensuring the integrity of data and services, decentralizing security operations, building trusted communication channels, controlling exchanges in IoT systems, and decentralizing operations of IoT systems. Figure 5 depicts the distribution of these categories among the reviewed papers.

Establishing trusted communication channels: Many IoT systems face the problem of establishing secure communication channels within or between them. These channels must be verifiable and non-repudiable to keep all parties accountable. Intra-system communications happen between IoT devices and their controllers. Inter-system interactions occur among IoT systems, as well as between them and cloud providers. Both forms of communication carry security risks as they tend to take place over insecure channels of wireless networks and Internet. As a result, neither IoT devices nor controllers can trust others entirely. Encryption and authentication alone are unable to secure the interaction if the counter-party misbehaves after the initial handshaking.

The reviewed BC-IoT systems leveraged BC networks as a trusted environment for intra- and inter-system com-

Table 3: Improvement objectives of BC-IoT systems in terms of new functionalities

Objective	Description	Frequency
Orchestrate processes involving IoT devices	Control the activities of devices within or across IoT systems in order to carry out predefined processes. These processes are not directly related to machine-to-machine trading.	**14**
Support the ownership and exchange of resources monitored by IoT systems	The most common forms of monitored resources are renewable energy and physical assets.	**12**
Maintain and conduct reputation assessment of devices and services	Use BC and smart contracts to store and update trust or reputation ratings of devices or services of IoT systems	**11**
Support the ownership and exchange of resources consumed by IoT systems	The most common forms of consumed resources are CPU time, storage space, and software patches.	9
Create provenance chain of IoT devices, data, or IoT-managed entities	Papers that create Supply-chain provenance, IoT hardware provenance, IoT data provenance fall into this category	7
Support the ownership and exchange of resources generated by IoT systems	The most common forms of generated resources are sensor data and actuation services.	6
Enable IoT service discovery	Facilitate the discovery of services offered by IoT devices via spatial or semantic matching	4
Synchronize and form consensus among IoT devices	Use BC as a pre-built mechanism to help devices within or across IoT systems achieve consensus.	2

munications. For instance, Ali et al. [27] used a Hyperledger Fabric network to exchange requests and responses between IoT systems. Liang et al. [52] maintained hashes of the exchanged commands and data between Unmanned aerial vehicles (UAV) and their controllers on a BC network to make the communication auditable.

Decentralizing Operation and Security of IoT Systems: Most IoT systems rely on trusted third parties on a cloud to operate and secure themselves. An advantage of this model is simplicity. Cloud services simplify infrastructure management as they can scale with the required amount of data and processing. They also simplify the control of distributed devices by providing a centralized perspective. For instance, users can control their smart home devices by interacting with cloud services instead of having to reach devices individually over the Internet.

The centralized model is not without disadvantages. Trust is the first issue, as this model trusts cloud service providers by default. However, this trust might not always be well-placed, as demonstrated by numerous security breaches by service providers in recent years. The second issue is availability. As cloud services are far from IoT devices, the connectivity between them is not guaranteed. In critical operations such as monitoring the vital signal of elderly people and sending alarms, disconnection can be fatal. Scalability is also an issue, as a few big cloud services might not be able to serve an entire expanding IoT, regardless of how elastic they are. Finally, cloud services also represent single points of failure, making IoT systems vulnerable to denial-of-service attacks.

Many BC-IoT systems address the problem of carrying out workflows and security processes of IoT systems without relying on a trusted third party or a centralized service provider. The reviewed BC-IoT systems either used BCs and smart contracts to replace cloud services entirely or as a mechanism to keep cloud services accountable [53]. The reviewed BC-IoT systems also migrated authentication, authorization, and trust management mechanisms from the cloud to BCs and smart contracts. BC networks provide a source of truth to operate security mechanisms. They can store access request [54, 55], cybersecurity incidences [47], reputation rating [43, 45], and white / blacklists of hosts [33]. As these records are immutable, they might serve as forensic evidence [56]. Some BC-IoT sys-

Table 4: Technical Problems of IoT Systems addressed by BC-IoT integration

Problem	Description	Frequency
Build Trusted Communication Channels		
Build intra-system communication channels	Establishing tamper-proof, accountable, and irreversible communication channels within IoT systems	14
Build inter-system communication channels	Establishing tamper-proof, accountable, and irreversible communication channels between IoT systems	6
Decentralize Security Operations of IoT Systems		
Decentralize the access control to and from devices	Operate access control to and from devices of one or many IoT systems without relying on a central access control manager	18
Decentralize the access control to data and services	Operate access control to data and services of one or many IoT systems without relying on a central access control manager	15
Decentralize the authentication of devices	Authenticate devices and users of IoT systems without relying on a central authentication manager	15
Decentralize the management of trust and reputation	Store and update reputation ratings of participants in IoT systems transparently and verifiably, without relying on a centralized service provider	12
Decentralize firmware delivery	Distribute security updates to IoT devices and control the update process without relying on a centralized infrastructure	6
Decentralize the management of identity and certificates	Store and update identities and digital certificates of participants in IoT systems without relying on a centralized service provider	3
Control and Incentivize Exchanges in IoT Systems		
Control and incentivize M2M trading	Provide incentives and control the business logic of exchanges between machines in an IoT ecosystem	21
Control and incentivize firmware distribution	Provide incentives and control the behaviour of volunteers that deliver security updates to IoT devices on behalf of the manufacturers	3

(To be continued)

tems use smart contract to run security logic based on the truth stored on-chain. For instance, smart contract can be used to can authorize access to and from devices [57], detect and prevent devices from participating in botnets [33], and update reputation rating of different participants in an IoT application [47]. An advantage of BC-based solutions is that they ensure integrity by cryptography and consensus protocols instead of human's assurance.

Controlling and incentivizing exchanges of resources in IoT systems: As the autonomy capability of IoT systems increases, more research considers the problem of machine-to-machine trading between these systems. Solving this problem requires means to record and direct the trades and the value storage vessels, such as currency units, to exchange. BC has been employed for its secure storage, the ability to run smart contracts, and the built-in support for cryptocurrencies. Renewable energy is the most common form of resources exchanged between IoT systems in smart grids (e.g., [22, 42, 37]). Another common type of exchanged resources are sensor data and services (e.g., [58, 59, 60, 61]).

This category also includes the problem of controlling and incentivizing the distribution of software updates for IoT devices. Currently, this distribution hinges on device manufacturers who pay the on-going operational costs. Thus, manufacturers might stop supporting older devices as a cost-cutting measure, even though these devices are still operating in the field. These devices, then, would become security risks to the entire IoT system. If a business shrinks, or that a manufacturer is out of business, the update delivery would cease. Reliance on a few manufacturers for a firmware delivery also presents a considerable

(Cont) Technical Problems of IoT Systems addressed by BC-IoT integration

Problem	Description	Frequency
Decentralize Operations of IoT Systems		
Decentralize the control of processes within an IoT system	Carry out non-financial processes that happen within IoT systems without relying on centralized controllers	11
Operate processes between IoT systems without intermediaries	Carry out non-financial processes that happen between IoT systems without relying on a trusted third party	5
Ensure the Integrity of Data and Services in IoT Systems		
Protect administrative and security records	Ensure the integrity of records that IoT systems use, such as access policies, device registries, and reputation ratings	30
Protect at-rest sensor data and event records	Ensure the integrity of sensor data and real-world events collected by IoT devices	26
Protect records of resource exchanges	Ensure the integrity of records regarding the ownership and exchange of assets related to the trading activities between participants in an IoT ecosystem	23
Protect indexes	Ensure the integrity of indexes pointing to data entry generated by IoT systems and stored elsewhere	9
Protect in-transit sensor data	Ensure the integrity of sensor data and real-world events collected by IoT devices when they are transmitted	2
Protect software instructions	Ensure the integrity of instructions for devices of IoT systems, such as flight plans	2
Protect configuration records of IoT devices	Ensure the integrity of records regarding the configurations of deployed IoT devices, such as bearing and zoom level of security cameras	1

Denial-of-Service risk. Decentralized delivery networks, built upon torrent or InterPlanetary File System (IPFS) protocols, is a potential solution. However, they might not be able to guarantee the integrity of the firmware. Moreover, they lack an incentivization model to encourage participants to host and deliver firmware updates.

BCs present a potential solution for these challenges. They offer integrity assurance via immutability. Cryptocurrency enabled by BCs can provide an incentive for participants to store and deliver firmware updates. For instance, based on these features, Leiba et al. [62] and Boudguida et al. [30] leveraged BC networks to build accountable firmware delivery networks. Leiba et al. placed firmware and smart contracts on an Ethereum BC. This contract releases fund whenever a firmware delivery has been fulfilled. Boudguida et al. pursued a similar approach but on a bitcoin-like BC.

Ensuring the integrity of devices, data, and services: Many IoT systems in the literature face the problem of ensuring the integrity of their data, devices, and services. BC has been employed to protect administrative and security records (e.g., [31, 63, 64, 65]), sensor data and event records (e.g., [24, 66, 32]), data indexes (e.g., [67, 68, 69]) resource exchange records (e.g., [22, 59, 70, 61, 48]).

Some BC-IoT systems address the problem of protecting physical devices against tampering. As IoT devices lack computing resources to run sophisticated security protocols and tend to be exposed to the environment, they are prone to tampering. Security breaches on IoT devices can have significant impacts. For instance, tampering camera settings in a smart city can lead to serious privacy breach and legal repercussion [71]. A solution to this problem is maintaining a record of devices configuration and check for deviations regularly. While centralized solutions exist, BC networks allow more transparency and allow for cross-organizational use cases. For example, Gallo et al. [71] maintains position, zoom level, and direction of view of cameras in a smart city. These configurations are derived directly from machine learning-based computer vision analysis on the video feed.

6. How do IoT Systems integrate BC?

In the previous section, we have discussed the "why" of BC integration in IoT systems. In this section, we discuss how the integration was carried out. As BC-IoT design is

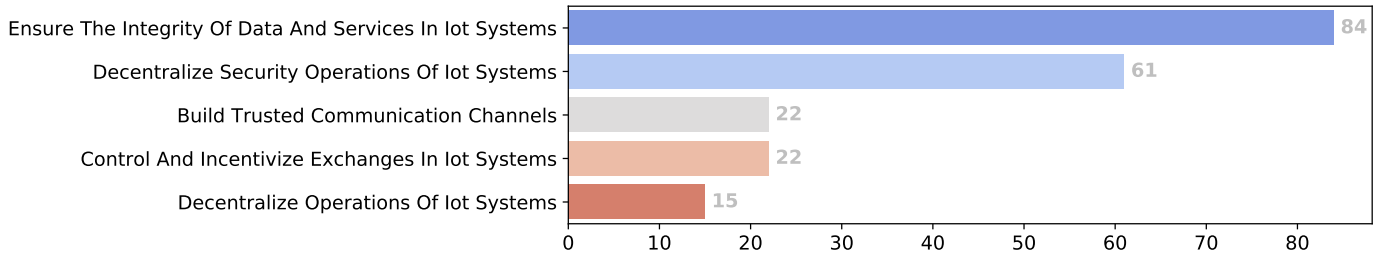

Figure 5: Distribution of problem categories.

a complex topic, it is inadequate and infeasible to capture it with a few noun phrases from one dimension without introducing significant ambiguity. Therefore, we capture the design of BC-IoT systems – the "how" of the integration – from three perspectives. First, we focus on where BC networks fit into IoT systems, physically and logically. Second, we look at the data and logic that IoT systems offload to their integrated BC. Third, we analyse the configurations of the integrated BC networks, such as the number of networks that they use, the permission model of those networks, their consensus protocols, and the platform on which they were developed. As these dimensions are not mutually exclusive but different perspectives of the same concept, they help to generate a rich picture of BC-IoT system design.

6.1. Positions of the integrated blockchains

Logical Position: We define the logical position of BC in an IoT system as the functional module of the system that BC adds or replaces. We identified these functional modules in a bottom-up manner by first extracting this information from individual studies and then reconcile the terminology. This approach moulds the findings to the reviewed studies instead of forcing them to fit a predefined reference architecture. Table 5 presents all of the identified logical positions of BC in IoT systems.

Access Control Manager, by far, is the most common role of BC in BC-IoT systems. This module assesses requests for accessing data and services based on predefined policies and grants access via tokens. It might also allows a party, such as a user, to delegate access rights to another, such as an IoT device [31]. Other security related roles of BC include Authentication Manager, Trust Management System, Public Key Infrastructure, and Device Integrity Verifier.

Another common role of BC in BC-IoT systems is Business Process Orchestrator. This module coordinates and records the activities related to the exchange of resources that IoT systems monitor, generate, or consume. In other words, they monitor and enforce contractual agreements between devices and clients (e.g.,[22, 38, 42, 58, 72, 59]). The other coordination role of BC is Workflow Orchestrator, which focuses primarily on cross-organization processes instead of asset trading (e.g., [25, 73, 74, 75, 49]).

BC also serves as secure storage and communication channels in BC-IoT systems. As Secure Data Store or Secure Index Store, BC holds sensing data and real-world events reported by IoT devices (e.g., [29, 56, 76, 77]) or maintains hash-pointers to data packets stored off-chain (e.g., [67, 68, 69]). As Intra-system communication channel, BC establishes verifiable links between devices and services within an IoT system (e.g., [51, 78, 79]). As inter-system communication channel, BC creates decentralized and accountable links between organizations to carry out processes such as supply chain tracking [80] or collaborative DDoS detection [81].

We have also identified some less common roles of BC. For instance, Fan et al., [82] proposed to use BC as the time source of IoT systems. This source provides the ground truth to synchronize the clock of the distributed IoT devices. Boudguiga et al., [30] and Lee et al., [36] proposed the use of BC to replace the existing centralized firmware delivery infrastructure of IoT systems. Niya, et al. [24] proposed to use BC to implement the event detector modules to harden them against tampering. These less common positions are outliers, which represent some interesting angles of BC-IoT integration.

Physical Position: The physical position of a BC network describes how its nodes are deployed on an Iot infrastructure. From the reviewed BC-IoT systems, we have identified nine physical positions of BC networks within IoT systems. For the sake of brevity, we introduce the following short-hand syntax to describe these deployment patterns of BC networks:

- The $[IoTnodetype]$ pattern denotes that the integrated BC consists of only one network, whose full nodes are deployed on $< IoTnodetype >$.
- The $[IoTnodetype]\&[IoTnodetype]$ pattern denotes that the integrated BC comprises at least two networks. Each $< IoTnodetype >$ specifies the deployment position of a network.
- The $[IoTnodetype]-Full-[IoTnodetype]-Lw$ pattern denotes that an integrated BC consists of only one network. This network contains both full nodes and lightweight nodes.
- Possible $[IoTnodetype]$ includes edge, fog, and cloud.

Table 5: Functional modules of IoT systems added or replaced by integrated blockchains.

Module	Description	Frequency
Access Control Manager	This component monitors and grants access to devices, data, and services of IoT systems based on policies specified by the owners	34
Business Process Orchestrator	This component is responsible for coordinating and recording activities of IoT devices, services, and human users to carry out business interactions, such as exchanging renewable energy, renting data, and delivering software updates	26
Secure Data Store	This component stores and facilitates the retrieval of data generated within IoT systems, such as sensing data and real-world events	21
Intra-system Communication Channel	This component allows devices and other components to exchange data and instructions within an IoT system	18
Authentication Manager	This component maintains identifying features of participants in an IoT systems and verifies their identity claims	15
Workflow Orchestrator	This component coordinates and records activities of devices and services within or across IoT systems to carry out predefined processes. These processes do not involve incentives and exchanges	14
Secure Index Store	This component maintains indexes pointing to data entries generated by IoT systems that are stored elsewhere	13
Trust Management System	This component handles the calculation and storage of trust or reputation scores of devices, users, and services of IoT systems	12
Inter-system Communication Channel	This component allows devices and other components to exchange data and instructions across IoT systems	8
Service Registry	This component maintains the details of services that an IoT system knows or offers	6
Update Delivery Infrastructure	This component distributes software and firmware updates from device manufacturers to deployed IoT devices	4
Service Matchmaker	This component matches requests with services known or offered by an IoT system	3
Digital Twins of IoT Devices	This component creates the digital representation of IoT devices, simplifying the interaction with them from software programs and services	3
Device Integrity Verifier	This component verifies the integrity of IoT devices, particularly their configurations, as they move across supply chains and operate in the field	2
Event Detector	This component detects events, such as violations of service level agreements, from the raw data recorded by IoT systems	2
Time Source	This component provides accurate time stamps for synchronizing devices and software programs of IoT systems	2
Public Key Infrastructure	This component handles the binding of public keys with identities of entities via the issuance of certificates	2

Table 6 presents the deployment patterns discovered from the reviewed BC-IoT systems.

The most prominent pattern is Cloud-based deployment. We consider an integrated BC network cloud-based if it is a public BC network (e.g., Ethereum), a Blockchain-as-a-service (e.g., IBM Block-chain Platform[3], Microsoft Azure Blockchain[4], Blockchain on AWS[5]), or deployed on an off-premise infrastructure. Ease of operation is the key advantage of this model because it removes infrastructure management from consideration.

The next common position of BC networks is on fog nodes, which are between clouds and sensors in both spatial location and computation capacity. For instance, Afanasev et al. [83] deployed a private Ethereum network across controllers of an industrial IoT system to control devices with smart contracts and maintain tamper-proof sensor data. Agrawal et al. [57] deployed a private Hyperledger Fabric BC network across fog nodes that manage different zones of a smart building to enable continuous authorization. Yang et al. [44] deployed a customized BC network across roadside units (RSU) to store and compute reputation ratings of smart vehicles. Fog deployment allows developers to modify the infrastructure parameters of a BC network, such as block rate and hash difficulty, to suit their IoT systems.

Edge-based deployment is less common among the reviewed BC-IoT systems. The reason might lie in the challenge of fitting BC nodes to edge devices with limited computation and storage capacity. One of such papers [45] deployed a customized BC network directly on onboard computers of vehicles to maintain reputation ratings of other vehicles. These ratings help vehicles to assess the trustworthiness of announcements coming from others without consulting cloud backends or fog nodes on the roadside.

Several BC-IoT systems utilized both full and lightweight nodes for their BC network. The most common configuration is having full nodes on a cloud and lightweight nodes on fog nodes. The purpose of these configurations is to bring BC networks closer to the edge without carrying over their resource demand. They keep the intensive mining operations on resource-rich nodes and bring only the ledger to the resource-constrained devices.

Finally, a few BC-IoT systems utilize more than one ledger [84, 36, 71, 69, 70, 85, 46, 86]. These systems contain more than one set of full nodes. Each ledger is hosted by a BC network which is maintained by a set of full nodes. By utilizing more than one ledger, the developers can fine-tune the integrated BC network to the demand and available resource of different parts of an IoT system.

6.2. Content of the integrated blockchains

The second perspective to describe an integrated BC network is their content – on-chain data and logic.

On-chain data: The data that a BC network stores can be encoded as transactions, stored in the transaction data, or kept in internal variables of smart contracts. Table 7 presents all of the identified on-chain data types in the reviewed BC-IoT systems.

Records of the authorization process lead as the most common on-chain data type. These records are either fed to BC-based access control managers or bi-product of their operation. Follow closely at the second position are records of resource exchanges which correspond to BC-based business process orchestrators. For example, Angin et al. [87] stored energy purchase requests on an Ethereum network. These records provide inputs to on-chain business logics as well as evidence of transactions. Huang et al. [59] used a BC network to store purchase requests for data and data access tokens. Xiong et al. [88] stored records of exchanging spare computing and storage resource of edge devices for monetary compensations. Sun et al. [41] operated a battery swapping service for electric vehicles with an Ethereum network. On-chain data comprises static and dynamic information of battery, as well as the value transferred between parties in a transaction.

Policies for access control and authentication are also common forms of on-chain data. For instance, Cha et al. [89] stored privacy policies and user preferences to control access to sensor data streams. Dukkipati et al. [55] encodes in BC transactions device registrations, access requests, and responses to be stored on BC. The BC also stores device identities and pointer to their XACML-based access control policies.

Other common on-chain data types are sensor readings and their hashes. These data types are generally reserved for high-stake, prone-to-tampering data. For instance, Niya et al. [24] used BC networks to store pollution sensing data, including Potential Hydrogen, turbidity, Carbon Monoxide, and Carbon Dioxide. BC ensures the integrity of these data, making them reliable evidence of pollution. Uddin et al. [90] stores physiological readings from body area sensors on BC to enable remote patient monitoring. Xie et al. [29] recorded the status of the surrounding environment of farm products as they moved through supply chains using IoT sensors and stored them on a BC. These records help to detect food safety issues.

Device and service interaction records are also common types of on-chain data. These records act as operation logs as well as evidence of misconducts. For instance, Ali et al. [27] used BC to maintain interaction records within crowd-based and autonomous enterprise networks, such as ride-sharing programs and supply chains. Dorri et al. [84] used BC to keep immutable records of interactions with smart home devices, such as requesting data. Hossain et al. [56] kept records of device-to-user, device-to-cloud, and device-to-device interactions on a BC network. Each record shows the taken action and the entities that involve. Investigators can use these records to find the responsible parties when IoT processes malfunction.

At the tail end of the distribution, we can find some un-

[3] https://www.ibm.com/blockchain/platform

[4] https://azure.microsoft.com/en-au/services/blockchain-service/

[5] https://aws.amazon.com/blockchain/

Table 6: Deployment patterns of the integrated blockchains.

Deployment Pattern	Description	Frequency
Cloud	The nodes of the blockchain network are deployed outside the operation area of IoT devices and their immediate processing nodes. This deployment mode covers the use of Infrastructure-as-a-Service, Blockchain-as-a-Service, as well as the use of existing public blockchain infrastructures	80
Fog	The nodes of the blockchain network are deployed on computing nodes that locate in the operation area of IoT devices, but at least one network hop away from the devices at the edge	20
Cloud & Fog	This mode of deployment consists of at least two blockchain networks. One follows the cloud-based deployment, the others are fog-based	6
Edge	The nodes of the blockchain network are deployed directly on the IoT devices at the edge of the network, which handle sensing and actuation	5
Cloud-Full – Fog-Lw	The deployed blockchain network utilizes both full and lightweight nodes. The full nodes are deployed on the cloud and might belong to a public blockchain network. The lightweight nodes are deployed on fog nodes	3
Cloud-Full – Edge-Lw	The deployed blockchain network utilizes both full and lightweight nodes. The full nodes are deployed on the cloud and might belong to a public blockchain network. The lightweight nodes are deployed on devices at the edge	3
Fog-Full – Edge-Lw	The deployed blockchain network utilizes both full and lightweight nodes. The full nodes are deployed on fog nodes. The lightweight nodes are deployed on devices at the edge	1
Cloud & Edge	This mode of deployment consists of at least two blockchain networks. One follows the cloud-based deployment, the others are edge-based	1
Fog & Edge	This mode of deployment consists of at least two blockchain networks. One follows the fog-based deployment, the others are edge-based	1

common on-chain data types. For instance, a BC network has been used to store SDN flow tables to for OpenFlow switches [51, 79], current time to synchronise decentralised IoT devices [82], and device configurations to verify their integrity [71].

On-chain logic: BC networks can execute users' programs as smart contracts. These programs accept transactions as inputs and generate the next state of the BC as outputs. This computation takes place in the mining process, and all full nodes can verify its outputs. Therefore, code execution via smart contracts is verifiable. The trade-off is that distributed applications based on smart contracts can be orders of magnitude slower than centralized applications.

Only about half of the reviewed BC-IoT systems utilized on-chain logic. Table 8 presents all the identified types on-chain logic in the reviewed BC-IoT systems. Contractual agreements between parties in M2M trading or update distribution are the most common form of on-chain logic. For instance, Missier et al. [58] utilized smart contracts to model and enforce the agreements between IoT data producers and consumers. These contracts monitor and detect issues in the flow of message between producers and consumers. By the end of each time window, they settle the payment automatically. Leiba et al. [62] used smart contract to automate the payment for firmware delivery. In this model, firmware distributors would acquire and install firmware updates on IoT devices in exchange for a proof-of-delivery. The smart contract then would exchange this proof-of-delivery for token awards.

On-chain logic has also been used to operate access control mechanisms, particularly the token-based variant. For instance, Ourad et al. [34] used smart contracts to imple-

Table 7: Types of on-chain data in the reviewed BC-IoT systems.

On-chain data type	Description	Frequency
Authorization Records	Records of requests for access and responses by the access control management component	28
Resource Exchange Records	Records of business interactions and exchanges between machines and users of IoT systems	26
Access Control Policies	Policies governing the access to devices, data, and services of IoT systems, such role-based policies, attribute-based policies, and lists of allowed and forbidden IP addresses	20
Sensor Data	Data and events collected by IoT devices	19
Sensor Data Hashes	Digests of data collected by IoT devices, which can be used as proof-of-existence or pointers to off-chain data packets	12
Service Interaction Records	Records of IoT systems invoking external services or of external parties invoking services offered by the systems	11
Device Descriptions and Identities	Records of identities, identifying features, and other descriptions of IoT devices, such as origin, configurations, hashes of their software agents, and whitelisted software.	11
Device Interaction Records	Records of interactions with devices in IoT systems	10
Reputation and Trust Ratings	Records of reputation or trust scores of devices and services participating in IoT systems	8
Service Descriptions	Metadata and descriptions of services offered or consumed by IoT systems	5
SDN Flow Tables	Flow tables for OpenFlow Switches in a software defined network	3
Data Indexes	Pointers to off-chain data packets	3
Instructions for Devices	Commands for IoT devices, such as flight plans and service invocations	2
Current Time	Time stamps provided by a trusted time source	2
Binaries of Updates	Binaries of software updates for IoT devices issued by their manufacturers	2
Authentication Records	Records of requests for authentication and responses by the authentication management component	2
Security Incident Records	Records of security incidents in an IoT system, such as failed authentications and unauthorized accesses	1
Hashes of Updates	Digests of software updates for proof-of-existence and integrity guarantee purpose	1
Processing Placement Records	Records of the offloading of computational tasks from devices to fog nodes or cloud services	1

ment an OAuth2-like protocol. These smart contracts authenticate users and broadcast access tokens to both users and their requested devices. These tokens are hashes of users' addresses, block mining time, and a random integer. Users would use these tokens to access devices, and devices would use these tokens to grant or reject the requests. Alphand et al. [91] also used smart contracts on an Ethereum network to implement token-based authorization. However, they embed in smart contracts sophisticated access rules.

At the tail end of the distribution, we found some uncommon types of on-chain logic. For instance, Niya et al. [24] implemented event detection logic with smart contracts. They used smart contracts to monitor on-chain environmental sensory data. Threshold violations would trigger and emit events, which remain immutable in the log of the BC network.

A few BC-IoT systems run additional logic on miner nodes to address some caveats of smart contracts. The most common use is to interact with off-chain actors, including users, devices, and cloud-services. Others offloaded computations that are too intensive and costly for on-chain execution. For instance, off-chain logic was used for cryptographic key generation [63, 28, 64, 92], reputation score calculation [93, 94, 44, 45], reasoning engine [95], and deriving devices' configurations from their readings [71].

6.3. Configurations of the integrated blockchains

The configuration is the third aspect to describe an integrated BC network. This aspect considers the number of BC networks that an IoT system uses, the permission type of these networks, the type of consensus that they use, and the BC technology used to build them.

Number of Integrated BC Networks: Most of the reviewed BC-IoT systems utilized only one BC network. This network is commonly an existing public network such as Ethereum. Some other BC-IoT systems deploy BC networks on their own Fog nodes and Cloud infrastructure.

Only 9 out of the 120 reviewed BC-IoT systems utilized more than one BC networks [53, 96, 84, 97, 69, 92, 85, 90, 46]. These systems combine faster private networks with a slower public network to increase performance while maintaining security. The private networks run on fog or edge nodes. They absorb the incoming data or provide localized services to sensors. Public BC networks interconnect and audit these private chains to secure the whole system.

Network Permission: A permission type defines who owns the right to access and append to a BC network. Three common permission types are public, private, and consortium.

A public BC is open to everyone. This permission type aligns with the original vision of BC as a ledger managed by a completely decentralized network. Transparency, auditability, and resilience are some of the advantages of public BCs. However, they require costly security measures. For instance, public BC networks adopt Proof-of-Work protocols to prevent Sybil attack. These protocols make voting expensive. They require participants to spend their resources to solve a puzzle and only append the submissions with the proper puzzle results. While these protocols increase security, they slow down public BCs consume a significant amount of computing resource. 7 out of the 10 reviewed systems used public permission type, making it the most common type.

Private and consortium permission types restrict access to a BC network. A single party controls a private network, while a group of entities controls a consortium network. These entities authenticate and authorize participants. They might also keep track of the identity of the participants. Therefore, costly proof-of-work protocols are generally unnecessary for these BC networks. Instead, they opt for less intensive consensus protocols that address only Byzantine Fault to speed up their processing and finalization.

Consensus Protocol: A consensus protocol includes the rules and the processes that BC nodes follow to maintain and validate a distributed ledger. For instance, this protocol specifies the syntax of transactions, the conditions that make a transaction valid. An essential part of consensus protocols is the mechanism to select a miner to append to the ledger. The Figure 6 depicts the distribution of consensus protocols among the reviewed BC-IoT systems.

An overwhelming number of the reviewed BC-IoT systems used some variants of Proof-of-Work (PoW) protocol, such as Ethereum's Dagger-Hashimoto protocol, Bitcoin's Nakamoto consensus protocol, or in-house developed protocols. In these protocols, network participants (i.e., "miners") race to find a number (a.k.a. "nonce") to make the block's hash smaller than a threshold value. The first miner to find the nonce earn the right to append its block to the chain and receive the mining rewards. The probability of a miner to win a mining round depends on its hash rate. PoW secures public BCs at the cost of significant energy consumption. Some of the reviewed BC-IoT systems modified PoW to address this issue. For instance, Uddin et al. [90] proposed selective PoW. This protocol removes the race between miners by selecting one miner to work at a time. Clients rank and select miners by a trust rating.

Practical Byzantine Fault Tolerance (pBFT) is an alternative to PoW. Byzantine fault is a condition in a distributed system where a component may fail, but the available information is imperfect to decide whether it has failed. pBFT protocol ensures that a BC network can reach consensus on the next state, despite Byzantine faults. It operates less than 1/3 of the nodes are malicious [98]. Different from PoW, pBFT does not demand costly puzzle-solving to counter Sybil attacks. Therefore, it is faster but less secure. Private and consortium chains such as Hyperledger Fabric tend to use pBFT. As these chains vet their participants, it is unlikely that 1/3 of nodes are malicious, and that Sybil attacks would occur.

Table 8: Types of on-chain logic in the reviewed BC-IoT systems.

On-chain logic type	**Description**	**Frequency**
No On-chain Logic	BC only serves as a mechanism for storing and sharing data. The IoT system does not offload logic to BC.	58
Resource Exchange Contracts	Specify and enforce terms, conditions, and agreements between users and machines of IoT systems that engage in resource exchanges, such as buying renewable energy, renting devices, compensating for software updates distribution, and distributing funds to machine's account. These contracts govern the change of ownership of assets recorded in the blockchain	27
Access Control	Assess incoming requests against existing policies, grant and manage access tokens, and update access policies	20
Workflow Orchestration	Control the activities of devices and services within or across IoT systems to carry out predefined workflow, such as a multi-party computation protocol or DDoS detection schemes	5
Service Matching	Matching requests with the existing services offered or consumed by IoT systems	4
Reputation Score Calculation	Calculate and update the reputation of trust ratings of devices and services of IoT systems based on their behaviours and assessment of their peers	4
Digital Twins of Devices	Maintain digital representatives of IoT devices, govern and update the state of devices based on incoming transactions	4
Data Index Maintenance	Update and maintain indexes of data packets stored off-chain	4
Integrity Checking	Verify the integrity of the input data, such as device instructions and configurations, against the existing records on-chain	2
Publish-Subscribe Protocol	Maintain a list of topics that users can subscribe to, and add messages to those topics based on incoming transactions	1
Authentication	Assess the incoming identity claims against the existing identifying records	1
QoS Calculation	Calculate quality metrics of industrial services based on the data reported by IoT sensors and submitted to the blockchain	1
Event Detection	Detect events, such as violations of service level agreements, based on the incoming sensing data	1
Identity Management	Update and manage the identity information of devices and services of IoT systems	1

Other alternatives rely on the hardware-enabled Trusted Execution Environment [69] (e.g., Software Guard Extensions (SGX) in Intel's CPUs), or a combination of PoW with Proof-of-Stake [85, 44]. Interestingly, some BC-IoT prototypes dropped consensus protocol entirely and relied on other mechanisms such as distributed trust assessment [84]. Merits of these approaches are up for debate. Detailed elaboration and comparison of BC consensus protocols are beyond the scope of this review. Interested readers can refer to Cachin, et al., [98] for a formalization and analysis of many prominent BC consensus protocols in practice.

BC platform: BC platform can be considered a bundle of pre-built design decisions to build a BC network. At a minimum, they allow running BC nodes and creating or participating in a BC network. Some BC platforms also support developing and deploying smart contracts. Figure 7 depicts the distribution of BC platforms used by the reviewed BC-IoT systems.

Ethereum[6] is the most common technology to build integrated BCs. Ethereum's wide adoption can be attributed to its early arrival, support for programmable smart contracts, and relatively matured technology stacks. Hyperledger Fabric[7] is also emerging in the research field. Its modular structure and support for private chains might be the key factors.

At the tail of the distribution, we have technologies such as Multichain[8], Monax and Eris[9], IOTA[10], and Hyperledger Iroha[11]. Finally, many works involve proprietary BC implementation. We classified them under the label of in-house BC systems.

We also extracted other features that characterize the internal architecture and operation of BC networks, including data structure of their ledger, model of their global state, and types of their smart contracts. We did not observe any deviation of these features from the norms that BC platforms establish. For instance, if a BC-IoT system uses Ethereum, then its BC generally uses a block-based data structure, uses accounts to model global state, and follows on-chain smart contract model. Thus, for the sake of conciseness, we would not elaborate further on these features.

7. How to optimize BC for IoT?

BC networks require a large amount of computing resource to operate and have low throughput. On the other hand, IoT systems have limited computing resource and generate data rapidly in a large volume. This section discusses the optimizations done by the reviewed BC-IoT systems to fit BC networks into the limitations of IoT systems.

Surprisingly, over half of the reviewed BC-IoT systems did not include any optimization. They either used BC technology as-is, relied on the existing public BC networks, or omit optimization details. For the systems that fine-tuned their integrated BC networks, their optimizations fit into three areas:

- Fitting a BC network to the limited resource offered by IoT infrastructure.
- Coping with the velocity and volume of IoT data coming from the sensing infrastructure.
- Verifying off-chain exchanges of resources.

Operating BC networks on resource-constrained IoT infrastructure: The first issue in this optimization area is hosting a resource-demanding BC on a resource-constrained infrastructure. The reviewed BC-IoT systems followed two approaches. The first one is migrating the whole BC to cloud-based virtual machines and connecting IoT devices directly to these nodes. From an IoT device's perspective, a BC backend is indistinguishable from a cloud back end [28]. The other approach is deploying lightweight BC nodes on resource-constrained IoT devices [99, 41, 74]. Lightweight nodes lack mining ability. They might hold only block headers instead of the entire transaction history. While these lightweight nodes can create and sign transactions, their ability to verify transactions is limited. They rely on full nodes for operating.

The second issue is allowing secured interaction between IoT devices and smart contracts when devices cannot run smart contracts themselves. Without the ability to run smart contracts, devices have to rely on the instructions from full nodes. This approach reintroduces trusted third parties and therefore is undesirable. Ellul et al. [100] proposed a split-virtual machine. They extended the Ethereum virtual machine with a part that runs directly on resource-constrained devices, removing the need for intermediaries.

The third issue is running resource-intensive security protocols that are necessary for privacy on BC networks. Stealth addressing protocols is an example. These protocols prevent linking and revealing the identity of account owners via transactions on BCs. Stealth addressing protocols are generally too demanding for low-powered IoT devices. Fan et al. [50] proposed a fast dual-key stealth address protocol. The static analysis shows that this optimized protocol can reduce overheads by at least 50% compared to the state-of-the-art.

Coping with the influx of IoT data: The first issue in this optimization area is handling the velocity of IoT data. The reviewed BC-IoT systems approached this problem from three angles. The first one is to make BC networks faster, meaning they take less time to process and finalize transactions. The reviewed BC-IoT systems hastened BC

[6] https://www.ethereum.org
[7] https://www.hyperledger.org/projects/fabric
[8] https://www.multichain.com
[9] https://monax.io
[10] http://iota.org
[11] https://www.hyperledger.org/projects/iroha

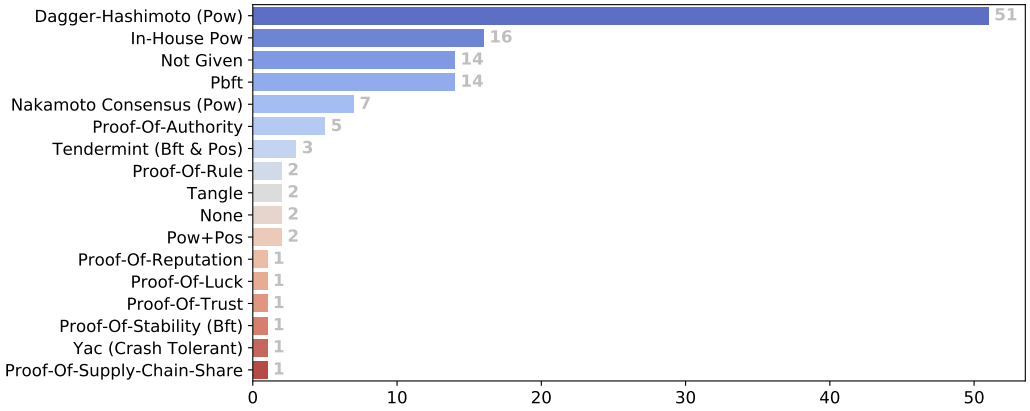

Figure 6: Distribution of consensus schemes in the reviewed BC-IoT prototypes.

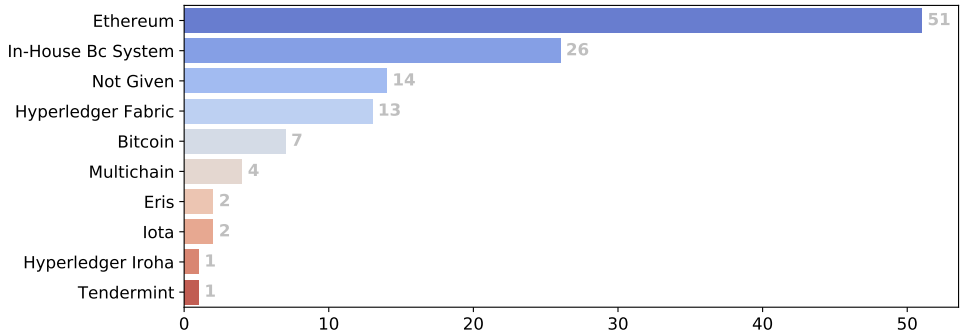

Figure 7: Blockchain platform employed by the reviewed BC-IoT systems

networks by making their consensus protocols faster and less costly. They altered the Proof-of-work protocol [90], replaced it with more efficient consensus protocols such as Proof-of-stake [37] and Proof-of-trust [69], or removed it altogether [84]. The second approach is to reduce the amount of data injecting into BC networks. The reviewed BC-IoT systems reduced IoT data by either aggregating [42] or filtering it [29]. The third approach is employing multiple BC networks [53, 101, 35]. Fast, private networks can absorb incoming traffic. Slower, public networks can coordinate and audit fast networks.

The second issue is taming the growth of ledger. One approach is offloading old transactions to external storage. This offload can be done by modifying the data structure of a BC to make the transaction log modifiable [26]. Another approach is to partition a BC network into clusters. Each cluster maintains only a subset of the BC network relevant to them. Qu et al. [78] applied hypergraph theory for this clustering task.

Verifying off-chain resource exchanges: As we discussed previously, orchestrating resource exchanges is one of the most common purposes of BC integration. These exchanges always include elements that happen off-chain, such as the delivery of firmware updates. Therefore, BC networks require mechanisms to monitor these off-chain exchanges.

The reviewed BC-IoT systems either modified the consensus protocol to motivate honest behaviors of service providers or introduced additional cryptography-based mechanisms. Leiba et al. [62] proposed proof-of-delivery to reward the delivery of firmware. A proof-of-delivery is a digital signature from IoT devices that firmware distributors can exchange for token rewards. This mechanism built upon a Zero-Knowledge Contingent Payment (ZKCP) protocol. Sharma et al. [85] proposed proof-of-service to link the delivery of service with the exchange of on-chain tokens. They employed a 2-hop consensus protocol that combines Proof-of-work and Proof-of-stake. The Proof-of-stake part is tied to service providers to motivate them to operate honestly. A side benefit of this 2-hop consensus is that it also helps protect BC networks in their early stage when they do not have adequate participation for proper decentralization.

8. Archetypes of BC-IoT Systems

Based on the improvement objectives, technical problems, positions and configurations of BC in IoT systems extracted from the analysed literature, we can synthesise the archetypes of BC-IoT systems. An archetype can be understood as the original pattern from which copies are made. Archetypes of BC-IoT systems, thus, embody the common patterns of usage and configuration of BC and

smart contracts within IoT systems that underlie most existing and emerging BC-IoT systems. It should be noted that these archetypes are not comprehensive recipes for BC-IoT systems because most of these combine multiple archetypes and make amendments to fit the particulars of their problems.

We identified 10 archetypes of BC-IoT systems from the literature as follows. For each archetype, we provide an overall description and details in terms of improvement objectives, technical problems, logical positions, on-chain data, and on-chain logic. We omit physical positions and configurations of BC as these features are not conclusive.

BC as the facilitator of M2M trading In this archetype, BC is used to record and direct the business processes that govern the exchange of resources between machines. These processes are generally autonomous. However, humans can influence them via setting the operation parameters or participate in the processes themselves. BC is leveraged for its guarantees on transparency, non-repudiation, and integrity of the data and processes involved in the exchanges.

- *Improvement Objectives:* Supporting the ownership and exchange of resources monitored, consumed, or generated by IoT systems; improving the overall security of the system in terms of accountability and non-repudiation.
- *Technical Problems*: Controlling and incentivizing M2M trading; protecting records of resource exchanges
- *Logical Position of BC:* Business process orchestrator
- *On-chain Data:* Resource exchange records
- *On-chain logic*: Resource exchange contracts

BC as the recorder of provenance information In this archetype, BC is used to improve the interoperability between IoT systems to enable the tracking of physical entities that moves across them. BC offers decentralised operation and integrity guarantee, which allow IoT systems to cooperate without trusting each other or a third party. While some BC-IoT systems of this archetype offload workflow logic to BC, most do not. Thus, we do not specify on-chain logic in this archetype.

- *Improvement Objectives:* Creating cross-organization provenance chain of device, data, or IoT-managed entities; Improving interoperability, accountability, and non-repudiation.
- *Technical Problems*: Building inter-system communication channels; protecting at-rest sensor data or event records
- *Logical Position of BC:* Intra-system communication channels; secure storage for data or indexes
- *On-chain Data:* Sensor data or hashes
- *On-chain logic*: Not utilized

BC as the access control manager for IoT systems In this archetype, BC is used to build decentralized access control mechanisms for either incoming or outgoing requests of IoT devices and services. These mechanisms can be used within an IoT system or shared by many. This archetype also covers collaborative whitelisting and blacklisting solutions. BC is leveraged for its decentralized operation and the ability to tamper-proof both the access policy and the access control process.

- *Improvement Objectives:* Improving integrity and confidentiality of the system
- *Technical Problems*: Decentralising access control to devices, data, or services of IoT systems; protecting administrative and security records such as access policies
- *Logical Position of BC:* Access control manager
- *On-chain Data:* Access control policies; authorization records
- *On-chain logic*: Access control

BC as the authentication manager for IoT systems This archetype uses BC to build decentralized authentication mechanisms for devices, services, or users. BC is leveraged for its decentralized operation, and the ability to tamper-proof both identifying information and authentication process. While some BC-IoT systems of this archetype offload authentication logic to BC, most utilize miners or other computing nodes to carry out the authentication off-chain based on-chain records. Thus, we do not specify on-chain logic in this archetype.

- *Improvement Objectives:* Improving authenticity of the system
- *Technical Problems*: Decentralising the authentication process; protecting administrative and security records such as identifying features of devices and services.
- *Logical Position of BC:* Authentication manager
- *On-chain Data:* Device descriptions and identities
- *On-chain logic*: Not utilized

BC as the trust rating manager for IoT systems This archetype uses BC as a secure storage for trust or reputation ratings of participants of IoT systems. Some BC-IoT systems of this archetype also use on-chain smart contracts for calculating trust ratings. However, most papers carry out this calculation off-chain. Therefore, we do not specify on-chain logic of this archetype.

- *Improvement Objectives:* Maintaining and conducting reputation assessment of devices and services in IoT systems; Improving integrity of the system
- *Technical Problems*: Decentralising the management of trust and reputation ratings; protecting administrative and security records such as trust and reputation ratings
- *Logical Position of BC:* Trust Management System
- *On-chain Data:* Reputation and Trust Ratings
- *On-chain logic*: Not utilized

BC as the update distribution infrastructure for IoT devices In this archetype, BC is used to maintain integrity of security patches for IoT devices and provide incentives to encourage the distribution of these patches in a peer-to-peer fashion. BC is leveraged for its integrity guarantee, non-repudiation, cryptocurrency, and smart contracts. Because the decentralized update delivery schemes generally involve volunteer peer nodes, these schemes generally include business process orchestration to direct the peers by controlling the incentives.

- *Improvement Objectives:* Supporting the ownership and exchange of resources consumed by IoT systems (i.e., security updates); improving accountability and non-repudiation of the system
- *Technical Problems*: Controlling and incentivising updates distribution; decentralizing updates delivery; protecting records of resource exchange
- *Logical Position of BC:* Update delivery infrastructure; business process orchestrator
- *On-chain Data:* Binary of hashes of updates; resource exchange records
- *On-chain logic*: Resource exchange contracts

BC as intra-system communication channels In this archetype, BC is leverated as a secure shared storage, through which devices within an IoT system communicate.

- *Improvement Objectives:* Improving security in terms of integrity, non-repudiation, and accountability
- *Technical Problems*: Building intra-system communication channels; decentralizing the control of processes within an IoT system
- *Logical Position of BC:* Intra-system communication channel; workflow orchestrator
- *On-chain Data:* Instruction for devices
- *On-chain logic*: Workflow orchestration

BC as inter-system communication channels In this archetype, BC is leverated as a secure shared storage that link devices in different trust domains. For instance, it can be used to coordinate distributed software agents in detecting DDoS attacks from IoT botnets.

- *Improvement Objectives:* Improving security in terms of integrity, non-repudiation, and accountability
- *Technical Problems*: Building inter-system communication channels; operating processes between IoT systems without intermediaries
- *Logical Position of BC:* Inter-system communication channel; workflow orchestrator
- *On-chain Data:* Instruction for devices; sensor data or hashes
- *On-chain logic*: Workflow orchestration

BC as a secure data storage for IoT systems In this archetype, BC acts as a tamper-proof database of data entries or pointers to off-chain data entries.

- *Improvement Objectives:* Improving security in terms of integrity
- *Technical Problems*: Protecting at-rest sensor data and event records or protecting indexes pointing to these data
- *Logical Position of BC:* Secure data store or index store
- *On-chain Data:* Sensor data, sensor data hashes, or data indexes
- *On-chain logic*: Not utilized

BC as a secure computing platform for IoT systems In this archetype, the ability to execute smart contracts of BC is leveraged to build a trusted computing platform upon untrusted computing nodes of IoT systems.

- *Improvement Objectives:* Improving security in terms of integrity
- *Technical Problems*: Protect software instructions
- *Logical Position of BC:* Workflow orchestrator
- *On-chain Data:* Varying depending on the use case
- *On-chain logic*: Workflow orchestration

9. Threats to Validity

Biases and incomplete selection of studies are two significant challenges of an SLR. Threats to the completeness of our review lie in the sources, keywords, exclusion criteria, and the time frame of the selection process. Source-wise, we chose Scopus as the primary abstract and index database due to its broad coverage and manual curation. Moreover, we complemented Scopus with the results from IEEE Xplore and ACM Digital Library to cover emerging venues in the domain of IoT and BC that are yet to be curated. Thus, the three chosen sources offer significant coverage over the computer science literature. However, our review might miss related studies in venues that are yet to be parts of the computer science research community or belong to a different field.

Keyword-wise, we used two relatively small sets of keywords to identify studies from the domain of BC and IoT. We derived these keywords from the research questions and fine-tuned them with pilot searches to minimize their false-positive rates. Some loosely related keywords were excluded to reduce the noise in the search results. We eliminated the term "distributed ledger" – an alternative name of BC – on the basis that "blockchain" has been mostly adopted as the canonical name of the concept. Thus, papers using the term distributed ledger would still mention blockchain and therefore, would be captured by our query. We also excluded terminologies that are not strictly synonymous with IoT. For instance, we excluded Cyber-Physical Systems, Ubiquitous Computing, and Embedded Systems because these systems are not necessarily IoT systems. Finally, we also excluded keywords denoting technical topics underlying BC, such as consensus protocol, distributed networking, and peer-to-peer as they are more fundamental than BC and thus match many papers that are not related to BC.

Regarding exclusion criteria, the decision to exclude grey literature such as technical reports and patents represents a threat to the paper selection's completeness. We acknowledge that grey literature plays an essential role in the canon of BC knowledge. For instance, the white paper of Bitcoin [18] and the yellow paper of Ethereum [21] are immensely influential yet have never been published as academic literature. However, as our goal is the Systematic Literature Review, we have limited our scope only on peer-reviewed research. A future Multivocal Literature Review (MLR) can be performed to complement the results of this SLR with the perspective of grey literature.

Regarding the time frame, we concluded the literature identification process in *April 2019*. This search time frame captures the papers published between 2015 to the end of 2018. Studies added to the databases after this point would not be considered in the review. This gap between the conclusion of the literature search and the reporting is an unavoidable limitation of SLRs. We mitigated this problem by targeting our study on the underlying structures of the BC-IoT domain – objectives, technical problems, and design. These structures would remain relatively stable, as new studies generally refine and improve the existing research branches rather than creating new directions.

Biases in an SLR can emerge from the human involvement in the process. We controlled the bias in our review in three ways. First, we minimize human involvement by automating many parts of the process. For instance, we developed a Python program to de-duplicate and combine results from three sources. Second, we conducted the paper selection in multiple rounds and maintained lists of selected and rejected papers in each round in a reference management system for validation purposes. Finally, we employed a cross-validation and adjustment process, in which the authors worked with a random sample of papers independently and used the consensus to adjust the selection and extraction process.

10. Summary and Concluding Remarks

Engineering IoT systems poses challenging questions regarding security, integrity, interoperability, and trust. The answers to these questions might lie in the immutability, auditability, and resilience offered by BC networks. In this paper, we have conducted a systematic study of 120 prominent BC-IoT systems published in the academic literature to form a picture of BC solutions for IoT systems. In this concluding section, we would present and elaborate on our key findings and discuss three arguments for and against the use of BC networks in IoT systems based on these findings. Finally, we would describe the short- and long-term research prospect on BC-IoT systems.

Why do IoT systems integrate BCs? We have investigated two perspectives: the objectives of integrating BC and IoT, and the technical problems that drive the integration. Objective-wise, eight out of the ten reviewed BC-IoT systems targeted quality improvement. Security improvement is the most common objective, and integrity is the most commonly targeted security quality. Other common qualities are reliability and performance. The reviewed BC-IoT systems leveraged the tamper-proof storage, trusted code execution, and the built-in redundancy of BC networks to address these qualities. Only two out of the ten reviewed BC-IoT systems used BC to provide new functionality to IoT systems. The most targeted functionality is enabling ownership and exchange of IoT resources, which is naturally inclined with the original purpose of BC networks.

Problem-wise, the reviewed BC-IoT systems targeted five problem categories. The most common ones are decentralizing operation and security of IoT systems. The reviewed BC-IoT systems used BCs either to replace cloud services or to keep them accountable. An advantage of BC-based solutions is that they ensure integrity by cryptography and consensus protocols instead of human's assurance.

How do IoT systems integrate BCs? We investigated the "how" of a BC integration from three aspects: the positions of BC networks in IoT systems, their content, and their configurations.

Position-wise, the top three functional modules of IoT systems that BC networks add or replace are business process orchestrator, authorization mechanism, and sensor data storage. Majority of the reviewed BC-IoT systems utilized only public BC networks or BC-as-a-service clouds. Several BC-IoT systems used both full and lightweight BC nodes to bring networks closer to edge without carrying over their resource demand. Only a few BC-IoT systems employed more than one ledger.

Content-wise, the most common on-chain data types are records of resource exchanges and interactions with devices and services. These records act as operation logs as well as evidence of misconducts. Other common on-chain data types are sensor readings and their hashes, which help to protect high-stake, prone-to-tampering data. At the tail of the distribution, we found some uncommon on-chain data types such as SDN flow tables, current time, and device configurations. On-chain logic was not as commonly used. Two prominent on-chain logic types are authorization mechanisms and contracts between providers and consumers.

Configuration-wise, a majority of the reviewed BC-IoT systems integrated only one BC network. This network is commonly a public, proof-of-work based network. Ethereum is the most common technology to build integrated BCs. We also investigated data structure, global state model of a chain, and types of smart contracts in the integrated BC networks. However, we did not detect any deviation from the norms of the BC technologies utilized by the reviewed BC-IoT systems.

What optimizations were performed on BCs to fit them into IoT infrastructure? Over half of the reviewed BC-IoT systems did not include any optimization. The remaining systems focused on fitting BC networks to the resource-constrained IoT infrastructure, to cope with velocity and volume of IoT data, and to verify off-chain exchanges of resources.

The reviewed BC-IoT systems followed two approaches to fit a resource-demanding BC on a resource-constrained infrastructure. The first one is migrating the whole BC to cloud-based virtual machines and connecting IoT devices directly to these nodes. The second approach is employing lightweight BC nodes on resource-constrained IoT devices.

The reviewed BC-IoT systems coped with the influx of IoT data in three ways. The first one is to reduce the time that a BC network takes to process and finalize transactions so that it can handle more transactions per second. The second approach is reducing the amount of data injecting into BC networks. The third approach is employing fast private networks to absorb incoming traffic while maintaining slower public networks for coordination and auditing.

To verify off-chain resource exchanges, the reviewed BC-IoT systems either modified the consensus protocol to motivate honest behaviors of service providers or introduced additional cryptography-based mechanisms.

10.1. A Case for BC: Decentralised Trusted Source of Truth

IoT systems require a trusted *source of truth* to coordinate their devices, fog nodes, cloud services, and other IoT systems. This truth represents what IoT systems consider the current facts, such as authorisation requests and responses [102, 34, 103, 54, 104], resource exchange records [22, 42, 58, 88, 105], trust ratings [43, 44, 45], and the current time [82]. Traditionally, an IoT system entrusts the truth to its cloud backend or an intermediary that orchestrate the interaction among systems. This entity has a global view of IoT systems and uses this view to maintain up-to-date truth. This centralized truth approach has some caveats. Single point-of-failure and reduced reaction time to external stimuli are some well-documented drawbacks. The other critical caveat is that this model *operates upon an assumed trust* which might not be guaranteed.

To trust another party is to believe that it would provide a service as per an agreement. A party is trusted when others consider it trustworthy enough to transact. IoT systems with centralized truth treat the truth-maintaining entity and the devices that provide inputs to the truth as trusted entities. While IoT systems might perform initial handshakes and frequent security checks, they still cannot be sure that their truth-maintaining entities and devices are trustworthy. The devices should not assume that commands, firmware, nor the truth that they perceive from the backend are entirely credible. The backend should not believe that IoT devices are honest either. Operating upon an assumed trust is dangerous.

BC networks can act as a decentralized trusted source of truth, which is auditable and guaranteed by mathematics. BCs can store the truth as the state of a ledger and transactions that drive the state changes. The trust in BC emerges from three factors. The first one is the cryptographic primitives that it uses, specifically public-key cryptography and digital signature. They guarantee that transactions were unaltered and came from the holder of the corresponding private key. They provide nonrepudiation and provenance. The second factor is the BC data structure, which embeds the hash of each transaction block into its subsequent block. Because a small change in the input leads to a substantial change in the hash, and because the hash of each block becomes a part of calculating the hash of the following block, any tampering would be apparent unless all the subsequent hashes are recalculated. However, if a node operates by itself, it would be able to recalculate these hashes to cover up the tampering, even if there is a PoW puzzle lock in each block. Therefore, the trustworthiness of BC hinges on the third factor: decentralization with complete redundancy.

A BC network is a collection of mistrustful participants. They trust neither the network nor their peers. In-

stead, they maintain a complete copy of the truth and verify everything coming their way: announcements, transactions, and blocks. As a result, a malicious entity can only overwrite the truth with the compromised one if it can control 51% of the network. This is considerably more difficult, given that every node in the network has its own agenda and vesting in the system. As the number of trustless nodes in a BC network increases, the network becomes resilient and the truth that it maintains become more trustworthy. Maintaining a decentralized trusted source of truth is by far the most common case for BC integration among the reviewed studies.

10.2. A Case for BC: Availability

In 2014, Jibo - the world's first family robot was announced with much fanfare, raising over $3 millions of crowdfund. It can greet parents, read to children, send reminders, deliver personal reports, and dance. Thus, Jibo was warmly welcomed to families when it finally arrived in 2017. Then in 2018, the company behind Jibo went bankrupt, and the servers supporting Jibo were shut down. This is just an example of the precarious nature of IoT systems whose brains live in remote cloud services. Their availability hangs on the survival and, to some degree, goodwill of service providers. Even if the service is still around, the availability is still not guaranteed, as the Internet connectivity of the devices might still be lost. This is also true in tactical systems and emergency response systems in disaster-struck areas where the communication has been knocked down.

BCs can increase the availability of IoT systems due to their complete redundancy. Each full node in BC network holds a complete copy of both data and logic, which means that there is not a single node that maintains the total control over data and logic of a system. If a node is lost, the remaining nodes in a BC can continue to function. In the case of Jibo, a BC hosted by dedicated volunteers might have been able to save a part of its brain so that it can continue to operate. If such chains can be established at the perimeter of IoT systems among their edge or fog nodes, then the systems might even be able to operate when the Internet connectivity is lost. Several of the reviewed BC-IoT systems considered availability a case for BC integration [30, 36, 106, 107, 33, 41].

10.3. A Case against BC: Performance and Scalability

Decentralized trusted truth and availability of BC come at a cost. *The first case against BC integration is performance.* BC networks rely on cryptographic primitives, data structure, and redundancy to provide decentralized trusted truth, All of these have negative impacts on the performance of a chain (i.e., throughput, latency, and bootstrap time [108]). In the case of Bitcoin, the maximum rate at which it can confirm transactions is 3.3 to 7 transactions per second [108]. It takes on average 10 minutes for a Bitcoin transaction to be included in a block, and 60 minutes for a transaction to be finalized [20]. Bootstrapping a BC full node is also a long process, clocking nearly four days in Bitcoin. Anecdotally, we observed a similar bootstrap time on Ethereum network when we set up a full node on a workstation with an 80 Mbps Ethernet connection.

In other words, *public BCs are generally slow. And costly.* As high as $6.2 USD per transaction confirmation in Bitcoin network [108].

This level of performance cannot keep up with the traditional payment systems and is vastly outpaced by the influx of IoT data. For instance, an IoT-based security camera can record up to 60 samples per second, while a microphone sensor can record from 8000 to more than 5 million samples per second. The reviewed papers proposed some solutions to bridge this performance gap:

- Reducing the data before committing to the chain [42, 29].
- Making the BC faster by altering its parameters and consensus protocols [84, 69, 37, 90].
- Using faster private chains to absorb the incoming traffic from IoT devices [53, 101].

The second case against BC integration is scalability. The complete redundancy which offers trust assurance and availability also means the ledgers on all full nodes are sizable and will grow without bound as an IoT system grows. Imagine we have a smart home that hosts a full node to run its automation logic. Even though the number of devices in our house does not increase, the software that runs our smart home would keep getting slower, and the data it requires would keep getting larger because more smart homes are brought online across the globe. This problem is exacerbated by the amount of data that IoT generates. Cisco estimates that by 2021, all people, machine, and things would produce nearly 850 zettabytes or 850 billion terabytes

Until these performance and scalability limitations are mitigated, practical BC integration in a production level might be limited.

10.4. Looking Forward: Faster Chains

Scaling BCs means finding the optimal compromise of the *Impossible Triangle: Security - Scalability - Decentralization.* Decentralization with complete redundancy is a key factor of the trustworthiness of BC networks. The more full nodes there are in a network to keep track of others, the more secure and anti-fragile a network becomes. But the more decentralized, the more redundancy is introduced into a network and the slower it becomes. The existing efforts to improve the performance of BC networks tends to lead to centralization, which might compromise their security. An example would be Ripple network, which replaces miners with 16 pre-selected verifiers that handle ledger updates.

BC scaling can be done in 2 layers. *Layer-1 scaling indicates the optimization done to the BC itself.* It is done by altering parameters of the BC network and changing its consensus protocols [84, 69, 37, 90]. In Ethereum, Layer-1 scaling is done by introducing Proof-of-Stake (PoS) via Casper protocol.

Layer-2 scaling indicates optimization done to the protocols built on top of BC networks, which can be created and modified without altering BC itself. There are three major approaches: side-chain, off-chain computation, and sharding. The side-chain approach involves employing faster, less secured chains to absorb the incoming transactions (e.g., [53, 101], Lightning Network payment protocol, and Ethereum's Plasma). A survey of side-chain technologies can be found in [86]. Off-chain computation approach involves offloading complex calculation off-chain in a way that is verifiable by the main-chain (e.g., TrueBit[12]). Sharding involves dividing data across multiple servers[13]. Layer-1 scaling might drive early innovation in public BC scaling; however, eventually, the public chain must be stabilized, and layer-2 scaling would become dominant[14].

10.5. Foreseeable Future: The Post-Quantum Cryptography World

By 2014, a quantum computer can factorize 56153 into its prime factors (233*241). While the number is by no means large, what is notable is that this factorization algorithm ran in polynomial time. As quantum computing continues to mature, it is not unreasonable to expect that in the foreseeable future, three hard mathematical problems underlying the current popular cryptography algorithms - integer factorization, discrete logarithm, and elliptic-curve discrete logarithm - would be solved. And by then, we would enter a post-quantum cryptography world in which the trusted cryptographic primitives might not protect our IoT system anymore.

One research direction relevant to BC-IoT systems is to secure a BC against the quantum attacks. Public-key cryptography is the most vulnerable. Proof-of-work would also be threatened by quantum computers. While it is true that the difficulty threshold of the BCs can adjust itself automatically to match with the available hash rate to ensure regular block time (10 minutes in case of bitcoin), quantum computers can still compromise public BCs by forcing centralization. Specifically, if some hypothetical superpowers have access to a functional quantum-based miner, they can drive the difficulty threshold so high that they effectively lock other miners out of a network and assume control of a chain. However, the risk to BC is not that severe, as its protocol can evolve quickly to replace the vulnerable cryptographic primitives with quantum-proof ones.

Low-power-long-living IoT devices along with legacy systems that IoT systems interact with, however, do not enjoy such luxury. They would be the most vulnerable, the weakest chains of IoT ecosystem in the post-quantum world. These points raise several questions: Can BC offer a decentralized security mechanism to protect these devices? Would BC evolve from keeping IoT cloud backends accountable for protecting the IoT ecosystem?

Or would a new technology emerge and take its place?

[12] https://truebit.io

[13] https://github.com/ethereum/wiki/wiki/Sharding-roadmap

[14] vitalik.ca/general/2018/08/26/layer_1.html

References

[1] D. Evans, The internet of things, Cisco IBSG White Paper.

[2] R. G. Brown, The corda platform: An introduction, Corda Whitepaper.

[3] A. Litan, Iot integration is a sweet spot for blockchain per gartner survey (2019).

[4] K. Christidis, M. Devetsikiotis, Blockchains and smart contracts for the internet of things, IEEE Access 4 (2016) 2292–2303. doi:10.1109/ACCESS.2016.2566339.

[5] N. Kshetri, Can blockchain strengthen the internet of things?, IT Professional 19 (4) (2017) 68–72. doi:10.1109/MITP.2017.3051335.

[6] K. Yeow, A. Gani, R. W. Ahmad, J. J. P. C. Rodrigues, K. Ko, Decentralized consensus for edge-centric internet of things: A review, taxonomy, and research issues, IEEE Access 6 (2017) 1513–1524. doi:10.1109/ACCESS.2017.2779263.

[7] M. Conoscenti, A. Vetro, J. C. De Martin, Blockchain for the internet of things: A systematic literature review, in: Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA. doi:10.1109/AICCSA.2016.7945805.

[8] A. Reyna, C. Martín, J. Chen, E. Soler, M. Díaz, On blockchain and its integration with iot. challenges and opportunities, Future Generation Computer Systems 88 (2018) 173–190. doi:10.1016/j.future.2018.05.046.

[9] A. Panarello, N. Tapas, G. Merlino, F. Longo, A. Puliafito, Blockchain and iot integration: A systematic survey, Sensors (Switzerland) 18 (8). doi:10.3390/s18082575.

[10] F. A. M., Blockchain technologies for the internet of things: Research issues and challenges.

[11] A. Dorri, Blockchain in internet of things: Challenges and solutions.

[12] S. K. Lo, Y. Liu, S. Y. Chia, X. Xu, Q. Lu, L. Zhu, H. Ning, Analysis of blockchain solutions for iot: A systematic literature review, IEEE Access 7 (2019) 58822–58835. doi:10.1109/ACCESS.2019.2914675.

[13] I. Makhdoom, M. Abolhasan, H. Abbas, W. Ni, Blockchain's adoption in iot: The challenges, and a way forward, Journal of Network and Computer Applications 125 (2019) 251–279.

[14] J. Sengupta, S. Ruj, S. D. Bit, A comprehensive survey on attacks, security issues and blockchain solutions for iot and iiot, Journal of Network and Computer Applications (2019) 102481.

[15] L. Atzori, A. Iera, G. Morabito, The internet of things: A survey, Computer Networks 54 (15) (2010) 2787–2805. doi:DOI10.1016/j.comnet.2010.05.010.

[16] P. Fremantle, A reference architecture for the internet of things, Technical Reportdoi:10.13140/RG.2.2.20158.89922.

[17] A. Narayanan, J. Clark, Bitcoin's academic pedigree, Communications of the ACM 60 (12) (2017) 36–45.

[18] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, Bitcoin Whitepaper.

[19] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. D. Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukoli, S. W. Cocco, J. Yellick, Hyperledger fabric: A distributed operating system for permissioned blockchains, in:

Proceedings of the Thirteenth EuroSys Conference, ACM, 3190538, 2018, pp. 1–15. doi:10.1145/3190508.3190538.

[20] X. Xu, C. Pautasso, L. Zhu, V. Gramoli, A. Ponomarev, A. B. Tran, S. Chen, The blockchain as a software connector, in: 2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA), 2016, pp. 182–191. doi:10.1109/WICSA.2016.21.

[21] G. Wood, Ethereum: A secure decentralized generalized transaction ledger, Ethereum Yellow Paper.

[22] Y. Zhang, J. Wen, An iot electric business model based on the protocol of bitcoin, in: Proceedings of the 18th International Conference on Intelligence in Next Generation Networks, ICIN 2015, pp. 184–191. doi:10.1109/ICIN.2015.7073830.

[23] B. A. Kitchenham, S. M. Charters, Guidelines for Performing Systematic Literature Reviews in Software Engineering, 2007

[24] S. R. Niya, S. S. Jha, T. Bocek, B. Stiller, Design and implementation of an automated and decentralized pollution monitoring system with blockchains, smart contracts, and lorawan, in: Proceedings of the IEEE/IFIP Network Operations and Management Symposium: Cognitive Management in a Cyber World, NOMS 2018, pp. 1–4. doi:10.1109/NOMS.2018.8406329.

[25] M. Samaniego, R. Deters, Using blockchain to push software-defined iot components onto edge hosts, in: ACM International Conference Proceeding Series. doi:10.1145/3010089.3016027

[26] R. C. Lunardi, R. A. Michelin, C. V. Neu, A. F. Zorzo, Distributed access control on iot ledger-based architecture, in: Proceedings of the IEEE/IFIP Network Operations and Management Symposium: Cognitive Management in a Cyber World, NOMS 2018, pp. 1–7. doi:10.1109/NOMS.2018.8406154.

[27] A. A. Ali, I. A. El-Dessouky, M. M. Abdallah, A. K. Nabih, The quest for fully smart autonomous business networks in iot platforms, in: ACM International Conference Proceeding Series, 2017, pp. 13–18. doi:10.1145/3178298.3178301.

[28] M. G. M. M. Hasan, A. Datta, M. A. Rahman, H. Shahriar, Chained of things: A secure and dependable design of autonomous vehicle services, in: Proceedings of the IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Vol. 02, pp. 498–503. doi:10.1109/COMPSAC.2018.10283.

[29] C. Xie, Y. Sun, H. Luo, Secured data storage scheme based on block chain for agricultural products tracking, in: Proceedings of the 3rd International Conference on Big Data Computing and Communications, BigCom 2017, pp. 45–50. doi:10.1109/BIGCOM.2017.43.

[30] A. Boudguiga, N. Bouzerna, L. Granboulan, A. Olivereau, F. Quesnel, A. Roger, R. Sirdey, Towards better availability and accountability for iot updates by means of a blockchain, in: Proceedings of the 2nd IEEE European Symposium on Security and Privacy Workshops, EuroS and PW 2017, 2017, pp. 50–58. doi:10.1109/EuroSPW.2017.50.

[31] Y. Kaga, M. Fujio, K. Naganuma, K. Takahashi, T. Murakami, T. Ohki, M. Nishigaki, A secure and practical signature scheme for blockchain based on biometrics, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 10701 LNCS, 2017, pp. 877–891. doi:10.1007/978-3-319-72359-4_55.

[32] L. Wu, X. Du, W. Wang, B. Lin, An out-of-band authentication scheme for internet of things using blockchain technology, in: Proceedings of the International Conference on Computing, Networking and Communications, ICNC 2018, pp. 769–773. doi:10.1109/ICCNC.2018.8390280.

[33] D. M. M. Mena, B. Yang, Blockchain-based whitelisting for consumer iot devices and home networks, in: Proceedings of the 19th Annual SIG Conference on Information Technology Education, International World Wide Web Conferences Steering Committee, 3241853, 2018, pp. 7–12. doi:10.1145/3241815.3241853.

[34] A. Z. Ourad, B. Belgacem, K. Salah, Using blockchain for iot access control and authentication management, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 10972 LNCS, 2018, pp. 150–164. doi:10.1007/978-3-319-94370-1_11.

[35] M. Samaniego, R. Deters, Zero-trust hierarchical management in iot, in: Proceedings of the IEEE International Congress on Internet of Things (ICIOT), pp. 88–95. doi:10.1109/ICIOT.2018.00019.

[36] B. Lee, J. H. Lee, Blockchain-based secure firmware update for embedded devices in an internet of things environment, Journal of Supercomputing 73 (3) (2017) 1152–1167. doi:10.1007/s11227-016-1870-0.

[37] C. Pop, T. Cioara, M. Antal, I. Anghel, I. Salomie, M. Bertoncini, Blockchain based decentralized management of demand response programs in smart energy grids, Sensors (Switzerland) 18 (1). doi:10.3390/s18010162.

[38] E. S. Kang, S. J. Pee, J. G. Song, J. W. Jang, A blockchain-based energy trading platform for smart homes in a microgrid, in: Proceedings of the 3rd International Conference on Computer and Communication Systems (ICCCS), pp. 472–476. doi:10.1109/CCOMS.2018.8463317.

[39] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, Y. Zhang, Consortium blockchain for secure energy trading in industrial internet of things, IEEE Transactions on Industrial Informatics 14 (8) (2018) 3690–3700. doi:10.1109/TII.2017.2786307.

[40] A. Pieroni, N. Scarpato, L. Di Nunzio, F. Fallucchi, M. Raso, Smarter city: Smart energy grid based on blockchain technology, International Journal on Advanced Science, Engineering and Information Technology 8 (1) (2018) 298–306. doi:10.18517/ijaseit.8.1.4954.

[41] H. Sun, S. Hua, E. Zhou, B. Pi, J. Sun, K. Yamashita, Using ethereum blockchain in internet of things: A solution for electric vehicle battery refueling, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 10974 LNCS, 2018, pp. 3–17. doi:10.1007/978-3-319-94478-4_1.

[42] T. Lundqvist, A. De Blanche, H. R. H. Andersson, Thing-to-thing electricity micro payments using blockchain technology, in: Proceedings of the Global Internet of Things Summit, GIoTS 2017. doi:10.1109/GIOTS.2017.8016254.

[43] M. Singh, S. Kim, Trust bit: Reward-based intelligent vehicle commination using blockchain paper, in: Proceedings of the IEEE World Forum on Internet of Things, WF-IoT 2018, Vol. 2018-January, pp. 62–67. doi:10.1109/WF-IoT.2018.8355227.

[44] Z. Yang, K. Yang, L. Lei, K. Zheng, V. C. M. Leung, Blockchain-based decentralized trust management in vehicular networks, IEEE Internet of Things Journal (2018) 1–1doi:10.1109/JIOT.2018.2836144.

[45] Z. Yang, K. Zheng, K. Yang, V. C. M. Leung, A blockchain-based reputation system for data credibility assessment in vehicular networks, in: Procedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC, Vol. 2017-October, pp. 1–5. doi:10.1109/PIMRC.2017.8292724.

[46] K. F., C. J., R. D., A. A., S. A., "a hardware-software codesign approach to identity, trust, and resilience for iot/cps at scale", in: "Proceedings - 2019 IEEE International Congress on Cybermatics: 12th IEEE International Conference on Internet of Things, 15th IEEE International Conference on Green Computing and Communications, 12th IEEE International Conference on Cyber, Physical and Social Computing and 5th IEEE International Conference on Smart Data, iThings/GreenCom/CPSCom/SmartData 2019", 2019. doi:10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00191.

[47] G. Spathoulas, A. Collen, P. Pandey, N. A. Nijdam, S. Katsikas, C. S. Kouzinopoulos, M. B. Moussa, K. M. Giannoutakis, K. Votis, D. Tzovaras, Towards reliable integrity in blacklisting: Facing malicious ips in ghost smart contracts, in: Proceedings of the Innovations in Intelligent Systems and Applications (INISTA), pp. 1–8. doi:10.1109/INISTA.2018.8466327.

[48] L. M., T. H., W. X., Mitigating routing misbehavior using blockchain-based distributed reputation management system for iot networks, in: "2019 IEEE International Conference on Communications Workshops, ICC Workshops 2019 - Proceedings". doi:10.1109/ICCW.2019.8757083.

[49] C. Y.-J., K. H.-J., L. I.-G., Scalable and secure internet of things connectivity, Electronics (Switzerland)doi:10.3390/electronics8070752.

[50] X. Fan, Faster dual-key stealth address for blockchain-based internet of things systems, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 10974 LNCS, 2018, pp. 127–138. doi:10.1007/978-3-319-94478-4_9.

[51] C. Qiu, F. R. Yu, H. Yao, C. Jiang, F. Xu, C. Zhao, Blockchain-based software-defined industrial internet of things: A dueling deep q-learning approach, IEEE Internet of Things Journal (2018) 1–1doi:10.1109/JIOT.2018.2871394.

[52] X. Liang, S. Shetty, D. Tosh, C. Kamhoua, K. Kwiat, L. Njilla, Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability, in: 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), IEEE, pp. 468–477.

[53] M. S. Ali, K. Dolui, F. Antonelli, Iot data privacy via blockchains and ipfs, in: ACM International Conference Proceeding Series, 2017. doi:10.1145/3131542.3131563.

[54] N. Tapas, G. Merlino, F. Longo, Blockchain-based iot-cloud authorization and delegation, in: Proceedings of the IEEE International Conference on Smart Computing, SMARTCOMP 2018, pp. 411–416. doi:10.1109/SMARTCOMP.2018.00038.

[55] C. Dukkipati, Y. Zhang, L. C. Cheng, Decentralized, blockchain based access control framework for the heterogeneous internet of things, in: ABAC 2018 - Proceedings of the 3rd ACM Workshop on Attribute-Based Access Control, Co-located with CODASPY 2018, Vol. 2018-January, 2018, pp. 61–69. doi:10.1145/3180457.3180458.

[56] M. Hossain, Y. Karim, R. Hasan, Fif-iot: A forensic investigation framework for iot using a public digital ledger, in: Proceedings of the IEEE International Congress on Internet of Things (ICIOT), pp. 33–40. doi:10.1109/ICIOT.2018.00012.

[57] R. Agrawal, P. Verma, R. Sonanis, U. Goel, A. De, S. A. Kondaveeti, S. Shekhar, Continuous security in iot using blockchain, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 6423–6427. doi:10.1109/ICASSP.2018.8462513.

[58] P. Missier, S. Bajoudah, A. Capossele, A. Gaglione, M. Nati, Mind my value: A decentralized infrastructure for fair and trusted iot data trading, in: ACM International Conference Proceeding Series. doi:10.1145/3131542.3131564.

[59] Z. Huang, X. Su, Y. Zhang, C. Shi, H. Zhang, L. Xie, A decentralized solution for iot data trusted exchange based-on blockchain, in: Proceedings of the 3rd IEEE International Conference on Computer and Communications, ICCC 2017, Vol 2018-January, pp. 1180–1184. doi:10.1109/CompComm.2017.8322729.

[60] D. A., G. D., J. P.P., N. A., Advancements towards global iot device discovery and integration, in: "Proceedings - 2019 IEEE International Congress on Internet of Things, ICIOT 2019 - Part of the 2019 IEEE World Congress on Services", 2019. doi:10.1109/ICIOT.2019.00034.

[61] P. H.-A., L. T.-K., P. T.-N.-M., N. H.-Q.-T., L. T.-V., Enhanced security of iot data sharing management by smart contracts and blockchain, in: "Proceedings - 2019 19th International Symposium on Communications and Information Technologies, ISCIT 2019". doi:10.1109/ISCIT.2019.8905219.

[62] O. Leiba, Y. Yitzchak, R. Bitton, A. Nadler, A. Shabtai, Incentivized delivery network of iot software updates based on trustless proof-of-distribution, in: Proceedings of the 3rd IEEE European Symposium on Security and Privacy Workshops, EURO S and PW 2018, pp. 29–39. doi:10.1109/EuroSPW.2018.00011.

[63] M. T. Hammi, P. Bellot, A. Serhrouchni, Bctrust: A decentralized authentication blockchain-based mechanism, in: Proceedings of IEEE Wireless Communications and Networking Conference, WCNC, Vol. 2018-April, 2018, pp. 1–6. doi:10.1109/WCNC.2018.8376948.

[64] R. Li, T. Song, B. Mei, H. Li, X. Cheng, L. Sun, Blockchain for large-scale internet of things data storage and protection, IEEE Transactions on Services Computingdoi:10.1109/TSC.2018.2853167.

[65] J. Zouari, M. Hamdi, T. Kom, Privacy preserving profile matching protocol for human-centric social internet of things, in: Proceedings of the IEEE 27th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), pp. 181–186. doi:10.1109/WETICE.2018.00042.

[66] M. A., R. A., K. B.H., H. L., Estimating service quality in industrial internet-of-things monitoring applications with blockchain, IEEE Accessdoi:10.1109/ACCESS.2019.2948269.

[67] B. Liu, X. L. Yu, S. Chen, X. Xu, L. Zhu, Blockchain based data integrity service framework for iot data, in: Proceedings of the IEEE 24th International Conference on Web Services, ICWS 2017, pp. 468–475. doi:10.1109/ICWS.2017.54.

[68] C. H. Lee, K. H. Kim, Implementation of iot system using block chain with authentication and data protection, in: Proceedings of the International Conference on Information Networking, Vol. 2018-January, pp. 936–940. doi:10.1109/ICOIN.2018.8343261.

[69] C. Machado, A. A. Frohlich, Iot data integrity verification for cyber-physical systems using blockchain, in: Proceedings of the 21st IEEE International Symposium on Real-Time Computing, ISORC 2018, pp. 83–90. doi:10.1109/ISORC.2018.00019.

[70] E. R. Sanseverino, M. L. D. Silvestre, P. Gallo, G. Zizzo, M. Ippolito, The blockchain in microgrids for transacting energy and attributing losses, in: Proceedings of the IEEE International Conference on Internet of Things, IEEE Green Computing and Communications, IEEE Cyber, Physical and Social Computing, IEEE Smart Data, iThings-GreenCom-CPSCom-SmartData 2017, Vol. 2018-January, pp. 925–930. doi:10.1109/iThings-GreenCom-CPSCom-SmartData.2017.142.

[71] P. Gallo, S. Pongnumkul, U. Q. Nguyen, Blocksee: Blockchain for iot video surveillance in smart cities, in: Proceedings of the IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC / ICPS Europe), 2018, pp. 1–6. doi:10.1109/EEEIC.2018.8493895.

[72] P. Danzi, A. E. Kalør, C. Stefanović, P. Popovski, Analysis of the communication traffic for blockchain synchronization of iot devices, in: IEEE International Conference on Communications, Vol. 2018-May, 2018. doi:10.1109/ICC.2018.8422485.

[73] C. Pahl, N. E. Ioini, S. Helmer, B. Lee, An architecture pattern for trusted orchestration in iot edge clouds, in: Proceedings of the 3rd International Conference on Fog and Mobile Edge Computing, FMEC 2018, pp. 63–70. doi:10.1109/FMEC.2018.8364046.

[74] L. Zhou, L. Wang, Y. Sun, P. Lv, Beekeeper: A blockchain-based iot system with secure storage and homomorphic computation, IEEE Access 6 (2018) 43472–43488. doi:10.1109/ACCESS.2018.2847632.

[75] J. A.P., L. E., S. M.Z., M. D., S. A., "rice donation system in orphanage based on internet of things, raspberry-pi, and blockchain", in: "Proceedings - 2018 4th International Conference on Computing, Engineering, and Design, ICCED 2018". doi:10.1109/ICCED.2018.00053.

[76] E. Kak, R. Orji, J. Pry, K. Sofranko, R. Lomotey, R. Deters, Privacy improvement architecture for iot, in: Proceedings of IEEE International Congress on Internet of Things (ICIOT), pp. 148–155. doi:10.1109/ICIOT.2018.00028.

[77] Z. Z., C. P., A. A., C. S., Parkchain: An iot parking service based on blockchain, in: "Proceedings - 15th Annual International Conference on Distributed Computing in Sensor Sys-

tems, DCOSS 2019". doi:10.1109/DCOSS.2019.00123.

[78] C. Qu, M. Tao, R. Yuan, A hypergraph-based blockchain model and application in internet of things-enabled smart homes, Sensors (Switzerland) 18 (9). doi:10.3390/s18092784.

[79] P. K. Sharma, S. Rathore, Y. Jeong, J. H. Park, Energy-efficient distributed network architecture for edge computing, IEEE Communications Magazine (2018) 2–9doi:10.1109/MCOM.2018.1700822.

[80] H. H., A. E., A. A., S. K., J. R., Smart contract-based approach for efficient shipment management, Computers and Industrial Engineeringdoi:10.1016/j.cie.2019.07.022.

[81] S. G., G. N., D. G.-P., T. G., Collaborative blockchain-based detection of distributed denial of service attacks based on internet of things botnets, Future Internetdoi:10.3390/fi11110226.

[82] K. Fan, S. Wang, Y. Ren, K. Yang, Z. Yan, H. Li, Y. Yang, Blockchain-based secure time protection scheme in iot, IEEE Internet of Things Journal (2018) 1–1doi:10.1109/JIOT.2018.2874222.

[83] M. Y. Afanasev, A. A. Krylova, S. A. Shorokhov, Y. V. Fedosov, A. S. Sidorenko, A design of cyber-physical production system prototype based on an ethereum private network, in: Proceedings of the 22nd Conference of Open Innovations Association (FRUCT), 2018, pp. 3–11. doi:10.23919/FRUCT.2018.8468296.

[84] A. Dorri, S. S. Kanhere, R. Jurdak, Towards an optimized blockchain for iot, in: Proceedings of the 2nd International Conference on Internet-of-Things Design and Implementation, IoTDI 2017 (part of CPS Week), 2017, pp. 173–178. doi:10.1145/3054977.3055003.

[85] P. K. Sharma, M. Y. Chen, J. H. Park, A software defined fog node based distributed blockchain cloud architecture for iot, IEEE Access 6 (2018) 115–124. doi:10.1109/ACCESS.2017.2757955.

[86] S. S., R. I.-H., M. W., K. M., C. G.H., Sh-blockcc: A secure and efficient internet of things smart home architecture based on cloud computing and blockchain technology, International Journal of Distributed Sensor Networksdoi:10.1177/1550147719844159.

[87] P. Angin, M. B. Mert, O. Mete, A. Ramazanli, K. Sarica, B. Gungoren, A blockchain-based decentralized security architecture for iot, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 10972 LNCS, 2018, pp. 3–18. doi:10.1007/978-3-319-94370-1_1.

[88] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, Z. Han, When mobile blockchain meets edge computing, IEEE Communications Magazine 56 (8) (2018) 33–39. doi:10.1109/MCOM.2018.1701095.

[89] S. C. Cha, J. F. Chen, C. Su, K. H. Yeh, A blockchain connected gateway for ble-based devices in the internet of things, IEEE Access 6 (2018) 24639–24649. doi:10.1109/ACCESS.2018.2799942.

[90] M. A. Uddin, A. Stranieri, I. Gondal, V. Balasubramanian, Continuous patient monitoring with a patient centric agent: A block architecture, IEEE Access 6 (2018) 32700–32726. doi:10.1109/ACCESS.2018.2846779.

[91] O. Alphand, M. Amoretti, T. Claeys, S. Dall'Asta, A. Duda, G. Ferrari, F. Rousseau, B. Tourancheau, L. Veltri, F. Zanichelli, Iotchain: A blockchain security architecture for the internet of things, in: Proceedings of the IEEE Wireless Communications and Networking Conference, WCNC, Vol. 2018-April, 2018, pp. 1–6. doi:10.1109/WCNC.2018.8377385.

[92] C. Qu, M. Tao, J. Zhang, X. Hong, R. Yuan, Blockchain based credibility verification method for iot entities, Security and Communication Networks 2018. doi:10.1155/2018/7817614.

[93] A. A. E. Kalam, A. Outchakoucht, H. Es-Samaali, Emergence-based access control: New approach to secure the internet of things, in: Proceedings of the 1st International Conference on Digital Tools and Uses Congress, ACM, 3240136, 2018, pp. 1–11. doi:10.1145/3240117.3240136.

[94] J. Kang, R. Yu, X. Huang, M. Wu, S. Maharjan, S. Xie, Y. Zhang, Blockchain for secure and efficient data sharing in vehicular edge computing and networks, IEEE Internet of Things Journal (2018) 1–1doi:10.1109/JIOT.2018.2875542.

[95] M. Ruta, F. Scioscia, S. Ieva, G. Capurso, A. Pinto, E. Di Sciascio, A blockchain infrastructure for the semantic web of things, in: Proceedings of the CEUR Workshop, Vol. 2161, p. 1DUMMY.

[96] V. Daza, R. Di Pietro, I. Klimek, M. Signorini, Connect: Contextual name discovery for blockchain-based services in the iot, in: Proceedings of IEEE International Conference on Communications, 2017. doi:10.1109/ICC.2017.7996641.

[97] R. Di Pietro, X. Salleras, M. Signorini, E. Waisbard, A blockchain-based trust system for the internet of things, in: Proceedings of ACM Symposium on Access Control Models and Technologies, SACMAT, 2018, pp. 77–83. doi:10.1145/3205977.3205993.

[98] C. Cachin, M. Vukolić, Blockchain consensus protocols in the wild, arXiv preprint arXiv:1707.01873.

[99] B. Y. Kim, S. S. Choi, J. W. Jang, Data managing and service exchanging on iot service platform based on blockchain with smart contract and spatial data processing, in: Proceedings of the 2018 International Conference on Information Science and System, ACM, 3209916, 2018, pp. 59–63. doi:10.1145/3209914.3209916.

[100] J. Ellul, G. J. Pace, Alkylvm: A virtual machine for smart contract blockchain connected internet of things, in: Proceedings of the 9th IFIP International Conference on New Technologies, Mobility and Security, NTMS 2018, Vol. 2018-January, 2018, pp. 1–4. doi:10.1109/NTMS.2018.8328732.

[101] S. Biswas, K. Sharif, F. Li, B. Nour, Y. Wang, A scalable blockchain framework for secure transactions in iot, IEEE Internet of Things Journal (2018) 1–1doi:10.1109/JIOT.2018.2874095.

[102] O. Novo, Blockchain meets iot: An architecture for scalable access management in iot, IEEE Internet of Things Journal 5 (2) (2018) 1184–1195. doi:10.1109/JIOT.2018.2812239.

[103] M. Saravanan, R. Shubha, A. M. Marks, V. Iyer, Smead: A secured mobile enabled assisting device for diabetics monitoring, in: Procedings of the 11th IEEE International Conference on Advanced Networks and Telecommunications Systems, ANTS 2017, pp. 1–6. doi:10.1109/ANTS.2017.8384099.

[104] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, J. Wan, Smart contract-based access control for the internet of things, IEEE Internet of Things Journaldoi:10.1109/JIOT.2018.2847705.

[105] J. Yang, Z. Lu, J. Wu, Smart-toy-edge-computing-oriented data exchange based on blockchain, Journal of Systems Architecture 87 (2018) 36–48. doi:10.1016/j.sysarc.2018.05.001.

[106] C. Li, L. J. Zhang, A blockchain based new secure multi-layer network model for internet of things, in: Proceedings of the 2nd IEEE International Congress on Internet of Things, ICIOT 2017, pp. 33–41. doi:10.1109/IEEE.ICIOT.2017.34.

[107] S. S. Choi, J. W. Burm, W. Sung, J. W. Jang, Y. J. Reo, A blockchain-based secure iot control scheme, in: Proceedings of the International Conference on Advances in Computing and Communication Engineering (ICACCE), 2018, pp. 74–78. doi:10.1109/ICACCE.2018.8441717.

[108] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. G. Sirer, On scaling decentralized blockchains, in: International Conference on Financial Cryptography and Data Security, Springer, pp. 106–125.