Blockchain and Smart Contract Solutions for Internet of Things Systems: A Systematic Review

***Abstract*—Internet of Things systems pose various technical problems, especially in the area of device management and security. Blockchain (BC) and Smart Contract (SC) systems have been integrated into IoT systems as solutions for the problems that they pose.**

In this paper, we present a systematic literature review on the integration of Blockchain and Smart Contract into IoT systems. We approach this topic from three perspectives: (i) the problems posed by IoT systems that call for BC and SC solutions, (ii) the specific way that BC and SC systems have been integrated into IoT systems, and (iii) optimisations done on BC and SC systems so that they can work with the constrained computing resources in IoT systems. Our review covers 90 research works. From these works, we identify real problems of IoT systems that have been addressed by BC and SC in these works. We also identify how BC and SC have been integrated into these IoT systems, as well as optimisation on individual components and architecture of a whole BC system so that it can work with IoT systems.

While prediction and ideas on how BC can be integrated and solve IoT problems have been discussed a lot in different mediums, from research writing to technical magazines; this paper is different in the sense that it derives these knowledge directly from the real works that have been conducted instead of prediction and anticipation. Comparing to some existing reviews on the topic, our review covers more perspectives, more articles, and also conduct in a more systematic way.

For IoT system engineers, this paper provides a catalogue of problems in IoT systems that have been addressed by BC and SC solutions. It also shows how the integration of BC and SC into IoT system has been conducted, and present some optimisation that has been conducted to enable such integration. For IoT researchers, this paper provides an overview of the landscape of IoT problems that have been addressed by BC technologies. Such overview helps to identify areas in IoT that would benefit from additional BC integration. For BC researchers, this paper provides an overview of how BC has been adapted and applied to IoT problems. Such an overview helps them to identify potential improvement so that BC can serve IoT systems better.

***Keywords*-Internet of Things, Blockchain, Smart Contract, Systematic Literature Review**

I. Introduction

Blockchain (BC) is a core component behind the emergence of cryptocurrency. This terminology can denote a distributed, potentially public, ledger which stores all transactions in a cryptographically-chained blocks [1], or the protocol that enables the distributed ledger. BC allows distributed peer-to-peer network among non-trusting members, in which they can interact with each other in a verifiable manner without relying on a trusted intermediary [2]. Due to these features, blockchain has been applied to logistics, smart cities [3], smart homes [4], vehicular networks [5], and more fields. In a tactical environment, blockchain allows distributed information sharing, immutable and auditable information, and consensus of information.

II. Background

In this literature review, we will discuss how Blockchain systems enhance or replace functional components of IoT systems to resolve their problems. We will also discuss how Blockchain systems are deployed in IoT systems and optimisations carried out on different functional components of Blockchain systems to make this integration happen. To prepare for these discussions, in this section we will present the necessary background. This includes some terminologies that we use to talk about IoT and BC systems, their functional components, and their deployment structure.

A. Internet of Things System

An Internet of Things system comprises Things (Devices) that report sensory data and physical events to generate Insights. These Insights drive Actions that improve businesses and processes [6], [7]. IoT systems enables "smart" applications that possess embedded intelligence and global awareness [8] to improve manufacture [**?**], logistic [**?**], retail [**?**], healthcare [**?**], city and building management [**?**], and home automation [**?**].

IoT systems monitor and control Physical Entities such as manufacturing machines, goods pallets, electrical grids, buildings, and rooms. IoT systems interact with Physical Entities indirectly via Virtual Entities such as software agents, services or data entries. These digital artefacts are synchronised with some properties of the corresponding Physical Entities, allowing changes an entity to reflect in the others [8].

IoT Devices synchronise Physical and Digital Entities. These devices are embedded into, attached to, or placed in the close vicinity of monitored Physical Entities. They provide technical interfaces to interact with or gain information from Physical Entities [8]. IoT Devices can be classified into three classes [9]: (i) 8-bit System-on-a-Chip controller with no OS, such as Arduino Uno, (ii) limited 32-bit systems based on Atheros and ARM chips, and (iii) 32- or 64-bit systems that can run full Linux or Android OS, such as Raspberry Pi 3 and BeagleBone.

IoT Devices host software components that provide data from, or used in the actuation on Physical Entities. These

software components are denoted as IoT Resources. These Resources are generally hardware dependent and heterogeneous. To simplify interactions with Resources, IoT systems can provide open, standardized interfaces to Resources and Devices called Services.

Quoting [Martin 2012]: IoT Services can be classified by their level of abstraction:

- Resource-level Services: exposes the functionality, usually of a Device, by accessing its hosted Resources.
- Virtual Entity level Services provides access to information at a Virtual Entity level.
- Integrated Services are the result of a Service composition of Resource-level or Virtual Entity level

In general, IoT systems involves a large number of endpoint devices whose resources, such as computing, memory, network bandwidth, and battery operation time, are severely limited. As a result, device management and security of these IoT systems raise a lot of problems.

For device management, IoT systems must be able to deliver software and security credentials update remotely to endpoint devices. They must be able to configure network parameters and physical capabilities of physical devices remotely. Finally, they must be able to disconnect rogue and stolen devices, as well as locating missing devices [9].

For security and privacy, IoT systems must handle initial registration of clients, including human users, software agents, as well as new IoT Devices. They must be able to provide anonymity and unlinkability to users. They must ensure that interactions can only happen between trusted parties. Finally, they must handle the establishment of integrity and confidentiality between parties with zero prior knowledge [8]. To make the matter more challenging, all of these tasks will be carried out on devices with very limited computing capabilities and operation time, due to battery life [9].

To support our discussion on the functionality that BC systems handle in an IoT system, and how BC systems are deployed in an IoT system, we will discuss a common architecture of IoT systems in this section.

As IoT system is a broad topic, we will look at it from two views: functional decomposition view which shows functional components of a general IoT system, deployment view which shows the mapping of functional components onto computing nodes of a general IoT system.

Knowledge presented in this section is adapted from the Architectural Reference Model in IoT-A [8]. This Reference Model extracts models, guidelines, best practice, architectural views and perspectives from IoT systems to guild the building of fully interoperable concrete IoT architecture and systems [8].

B. Functional-decomposition of IoT Systems

Figure 1 depicts the functional components of an IoT System defined in IoT-A Reference Architecture [8]. IoT Systems bridge between IoT Devices and Applications. They include 23 functional components clustered into 7 function groups. It should be noted that a concrete IoT system might not include all components specified in this reference architecture.

The Communication function group contains communication schemes and technologies to interact with IoT Devices and interact with IoT Services. The IoT Service and Virtual Entity function groups contain services to interact with data and functionality on the IoT Resources and Virtual Entity levels, respectively. They also contain functionality to discover, look up, and resolve service name for service locator. The Service Organisation function group handles the composition and orchestration of services at different level of abstraction to satisfy requests from the clients. The IoT Process Management function group handles the integration between business processes and IoT systems. In other word, it integrates IoT Resources and Services into workflows of business processes.

Security function group handles the security of the system, privacy of users, and trust of involving parties.

- The Authorization function component is the front end for managing policies and performing access control decisions based on access control policies. In other words, it has two functionality: to determine whether an action is authorized based on policies, and to manage policies themselves.
- The Authentication functional component involves in the authentication of users and services. It checks the authenticity of credentials provided by a user and return an assertion that is required to use IoT and VE Services.
- The Identity Management functional component creates fictional identity and related security credentials for users and services to protect their privacy.
- The Key Exchange Management (KEM) functional component enables secured communication between two or more users or services that do not have initial knowledge of each other. It distributes keys in a secure way and register security capabilities of different parties that want to be mediated by KEM.
- The Trust and Reputation Architecture functional component collects user reputation scores and calculate services trust levels.

Management function group handles the device management functionality. It consists of five functional components.

- The Configuration functional component is responsible for initialising the system configuration and tracking configuration changes. It retrieves configurations of a system either from the history or from the current system and tracking the change of configuration. It also handle setting configuration when initialising or changing the system configuration.
- The Fault functional component is to identify, isolate, correct, and log fault that occurs in an IoT system.
- The Member functional component is responsible for storing information about entities that belong to the system. Such information includes their ownership, capabilities, rules, and rights. It is responsible for continuous monitoring of members. It allows retrieving members of

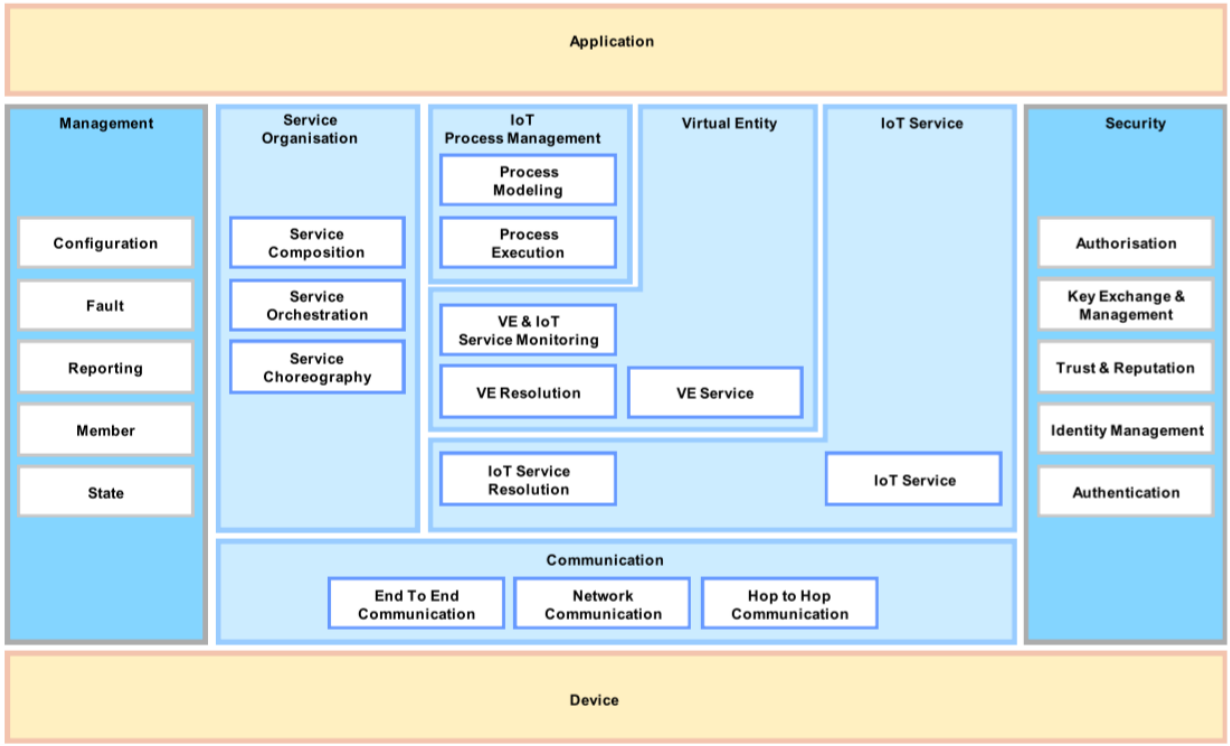

Fig. 1. 23 Functional Components of an IoT System organised into 7 Function Groups. Reproduced from IoT-A Reference Architecture for IoT Systems [8]

a system based on certain filters. Finally, it handles the update of membership information such as to add new member or unregister members from the database.

- The Reporting functional component helps determine the efficiency of the current IoT system. The information from this functional component helps to predict future issues, as well as for billing tasks.
- The State functional component monitors and predicts state of the IoT system. This component has three functions. The first one is to enforce a certain state change of the system. The second function is to monitor the state of the system. The third function is to predict the state of a system at a given time, based on the state history.

C. Deployment view of IoT Systems

Deployment view shows how non-software components of a software system, such as computer nodes, are arranged and interact with each other. It also show how software components are mapped onto hardware components. Figure 2 depicts the deployment view of IoT systems that we will follow in this review. This deployment view is adapted from Intel's IoT Reference Architecture [6]. In this section, we discuss only about common types of hardware nodes in an IoT system, how they communicate, and some options to map software components on those hardware nodes.

Hardware components in an IoT systems are either on- or off-premise. On-premise components include sensors and actuators, low- or high-powered embedded computers acting as controllers of sensors and actuators, and gateway devices.

Sensors convert environmental features and events into digital signal. Actuators modify the physical environment based on incoming digital signal. Together with controllers, sensors and actuators form Devices that bridge Physical Entities with Digital Entities. IoT systems utilise either low power 8-bit controllers that offer limited OS support, or high power 32- or 64-bit platforms that support full Linux or Android OS. Certain software components of IoT systems can be deployed directly on these high-powered controllers. In this review, we call this form of deployment the Edge Deployment.

More IoT systems include even more capable computing nodes on-premise, called gateways or fog nodes. These gateways are generally powered by electrical line, giving them unlimited operation time. These gateways can act as intermediate to connect low-powered IoT devices to the Internet. They also provide data aggregation service, perform real-time analytics, run machine learned models, as well as securing and managing on-premise devices. Certain software components of IoT systems can be deployed on these gateways. In this review, we call this form of deployment the Gateway Deployment.

Finally, most IoT systems contain a remote backend, in forms of data centres or cloud services. The backend of IoT systems has significant processing and storage capability, making them suitable for storing and processing Big Data generated by IoT devices [7]. The downside of backend is delay due to network communication, making them unsuitable for real-time analytics and responses. Certain software components of IoT systems can be deployed directly on these

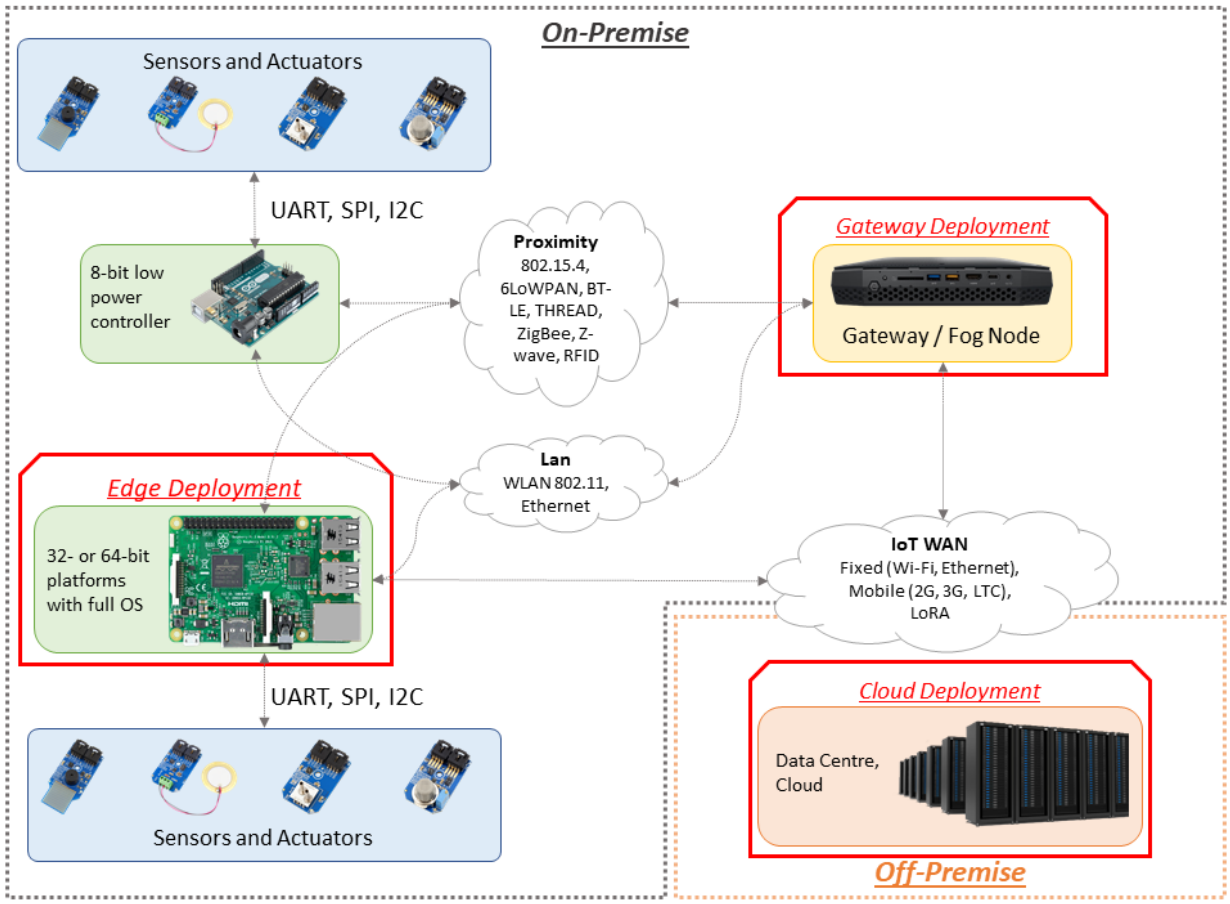

Fig. 2. Deployment view of IoT Systems, adapted from Intel's IoT Reference Architecture [6]. Three deployment options (Edge, Gateway, Cloud) are highlighted.

gateways. In this review, we call this form of deployment the Cloud Deployment.

IoT systems, in general, mix and match three forms of deployment. How they mix depends on many factors, including their need for real-time reaction.

III. Review Protocol

A. Background

A systematic review is a trustworthy, rigorous, and auditable methodology for evaluating and interpreting all available researches that are relevant to a research question, a topic area, or a phenomenon [10]. Different from an unstructured literature review, which is sometimes referred to as a narrative review, a systematic literature review (SLR) revolves around a review protocol which specifies the research questions being addressed, search strategy to detect relevant literature, explicit criteria for including and excluding studies from the review, and the information to be extracted from each study. As a result of this rigorous conduct, an SLR can synthesize new insights in a scientific manner.

In the following section, we will present some details of the review protocol that we defined for this project.

B. Research Questions

The problem that this research addresses is determining the problem space posed by IoT systems, and how BC technologies have been modified and applied to address these problem? This research problem is decomposed into *three research questions (RQ)* to guide the literature review process.

RQ1: What are problems of IoT that have been addressed by Blockchain technology?

- Potential sub-questions: How do actual problems compare to anticipated problems? Do they align with each other? Is there any anticipated problem that has not been addressed by existing BC research? Vice versa, is there any real problems that have not been anticipated?

RQ2: How have Blockchain technology been used to address problems identified in RQ1?

- RQ2.1: How do Blockchain and Smart Contract fit into an IoT system?
- RQ2.2: How has the Blockchain - Smart Contract system been designed and implemented?

RQ3: How have BC technology been adapted to use in IoT systems? This question concerns with optimisations that has

been carried out to make BC compatible with an IoT system.

C. Study Selection Protocol

Study Selection Process: We will apply the following process to select studies systematically to answer our research questions.

1) Retrieve potential studies by querying digital libraries. Cross-validation might be carried out to adjust the query.
2) Remove duplicates and irrelevant entries from the query results, such as conference proceedings.
3) Combine results from queried sources to create a set of potential studies.
4) Apply selection criteria on title and abstract of potential studies.
5) Apply selection criteria on the full text of potential studies.
6) Cross-validation: Send a random sample of selected and rejected articles to co-investigators for repeating the selection and comparing the selection result.

Sources of Studies: We will conduct our query for relevant studies on three digital libraries: ACM Digital Library[1], IEEE Xplore[2], and Scopus[3]. We refrained from using Google Scholar as a source because it lacks the structured query capability and the reproducibility of other digital libraries.

Query: Studies that we aim to include in our review lie at the intersection between Blockchain and Internet of Things research. Based on this requirement, we constructed the following query to retrieve potential studies from our chosen sources:

```
[``Blockchain'' OR ``block chain'']
                AND
[``Internet of Things'' OR ``IoT'' OR
   ``Web of Things'' OR ``WoT'' OR
 ``Industrial Internet of Things'']
```

Query results will be filtered by their field of study and language. We will assess only studies in Computer Science, Engineering, and Mathematics, which were written in English.

Selection Criteria We will assess each potential study against the following four selection criteria:

- *Criterion 1:* Include only research and engineering work. This criterion corresponds to the purpose of our review: to derive insights on problem space and solution space of BC-integrated IoT systems from actual conducted research and engineering works. According to this criterion, we will reject all secondary studies, short and position papers, and primary studies that do not include implementation and evaluation details.
- *Criterion 2:* Include works that describe a specific mechanical or technique to apply BC in solving an IoT problem. This criterion corresponds to RQ1 and RQ2.
- *Criterion 3:* When an author or a group of authors create a set of publications on the same system, only the latest and most comprehensive paper is included.
- *Criterion 4:* Include works that adapt or optimise elements of BC, such as architecture, consensus mechanism, and mining, to make it suitable for IoT uses. This criterion corresponds to RQ3.

[1]https://dl.acm.org/
[2]https://ieeexplore.ieee.org/
[3]https://www.scopus.com/

D. Data Extraction Protocol

Data Extraction Conduct and Cross-Validation: The extraction of data from selected studies to answer the research questions is guided by variables and levels. A variable captures data from studies on a feature that we attempt to quantify or qualify to answer the research questions. Levels of a variable are its possible values. More details on the proposed variables and levels will be presented in the following section.

For each selected study, we will extract the data corresponding to the variables and store it in a database. For cross-validation purpose, a random sample of articles will be sent to co-investigators for them

to perform the data extraction. Their extracted data will then be compared with the data extracted by the primary investigator. Such a process helps reducing the subjectivity and bias.

Variables and Levels

Variables of RQ1: These variables extract the data for establishing the problem space of IoT systems for which BC-based solutions have been proposed.

- IoT problem that the study aims to solve.

Variables of RQ2: To answer the research question RQ2.1 on how BC-SC system fits into an IoT system, we extract:

- Functional module of an IoT system that has been replaced or added by a BC-SC system.
- Types of BC-SC nodes (i.e., miner, full, light-weight, web client) that were deployed on computing nodes of an IoT system (i.e., cloud, fog, edge)

To answer the research question RQ2.2 on how BC-SC system has been designed and implemented, we extracted data relating to four perspectives:

- Functional decomposition of an BC-SC system
- Ledger and Smart Contract
- Access control
- Implementation

Figure 3 depicts the features to be extracted.

Variables of RQ3: These variables extract the data on optimizations to adapt BC to the nature of IoT systems.

- What is the perceived challenge of applying BC in IoT systems?
- What solution was proposed to address such perceived challenges?

Levels of these variables will be defined and fine-tuned when we conduct the data extraction.

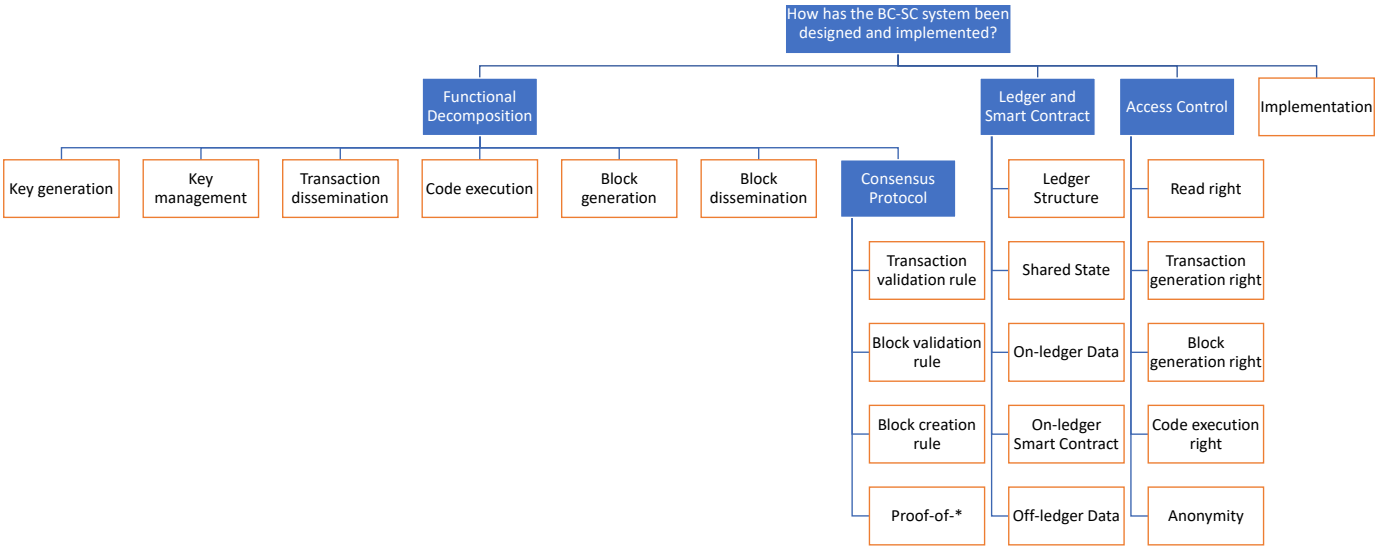

Fig. 3. Extracted features to describe the design and implementation of a BC-SC system.

E. Synthesis Protocol

In the synthesis phase of an SLR, new insights are generated from the extracted data to answer the posed research questions, identify gaps, and suggest new research directions.

Synthesis for RQ1: The purpose of RQ1 is forming a problem space of BC-IoT systems. To address RQ1, we will carry out two activities.

First, we will construct a taxonomy of IoT problems that have been addressed by BC-based techniques and mechanisms. This taxonomy must not be as arbitrary as application domains. For example, existing works [11], [**?**] have classified BC solutions into domains such as smart homes, smart building, smart city. This classification is ambiguous and offer little benefits. A potential alternative is classifying the identified problems as non-functional requirements and software qualities.

Second, we can quantify the distribution of BC-IoT works across IoT problems. This information will allow IoT researchers to identify under-addressed areas.

Synthesis for RQ2 and RQ3: Tabulation and visualisation of the distributions of variables related to these questions would be adequate. The primary challenge would lie in drawing conclusions from these numbers, not the numbers themselves.

IV. Extracted Data

A. Problems of IoT Systems

Figure 4. Figure 5.

V. Discussions

VI. Related Works

VII. Conclusion

References

[1] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Proceedings - 2017 IEEE 6th International Congress on Big Data, BigData Congress 2017*, pp. 557–564.

[2] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[3] S. Huckle, R. Bhattacharya, M. White, and N. Beloff, "Internet of things, blockchain and shared economy applications," in *Procedia Computer Science*, vol. 58, pp. 461–466.

[4] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for iot security and privacy: The case study of a smart home," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2017*, pp. 618–623.

[5] P. K. Sharma, S. Y. Moon, and J. H. Park, "Block-vn: A distributed blockchain based vehicular network architecture in smart city," *Journal of Information Processing Systems*, vol. 13, no. 1, pp. 184–195, 2017.

[6] Intel, "The intel iot platform: Architecture specification white paper," *Technical Report*, 2015.

[7] Microsoft, "Microsoft azure iot reference architecture v2.1," *Technical Report*, 2018.

[8] M. Bauer, M. Boussard, N. Bui, F. Carrez, Jardak, L. De, Magerkurth, S. Meissner, Nettstrter, A. Olivereau, Thoma, W. Joachim, Stefa, and A. Salinas, *Internet of Things Architecture IoT-A Deliverable D1.5 Final architectural reference model for the IoT v3.0.* Technical Report, 2013.

[9] P. Fremantle, "A reference architecture for the internet of things," *Technical Report*, 2015.

[10] B. A. Kitchenham and S. M. Charters, *Guidelines for performing systematic literature reviews in software engineering*. 2007.

[11] A. Panarello, N. Tapas, G. Merlino, F. Longo, and A. Puliafito, "Blockchain and iot integration: A systematic survey," *Sensors (Switzerland)*, vol. 18, no. 8, 2018.

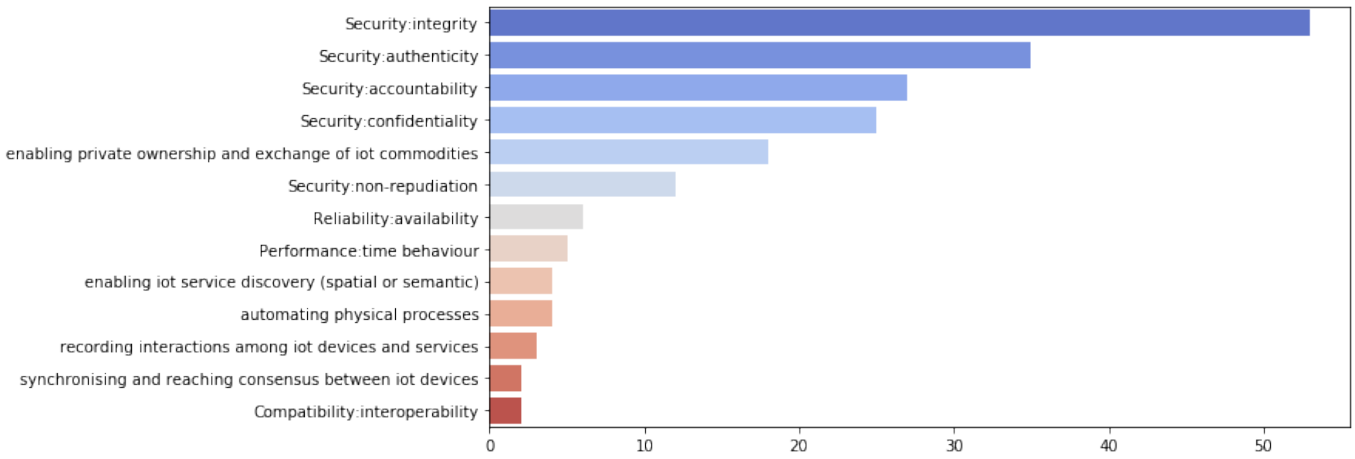

Fig. 4. Distribution of problem areas addressed by BC integration.

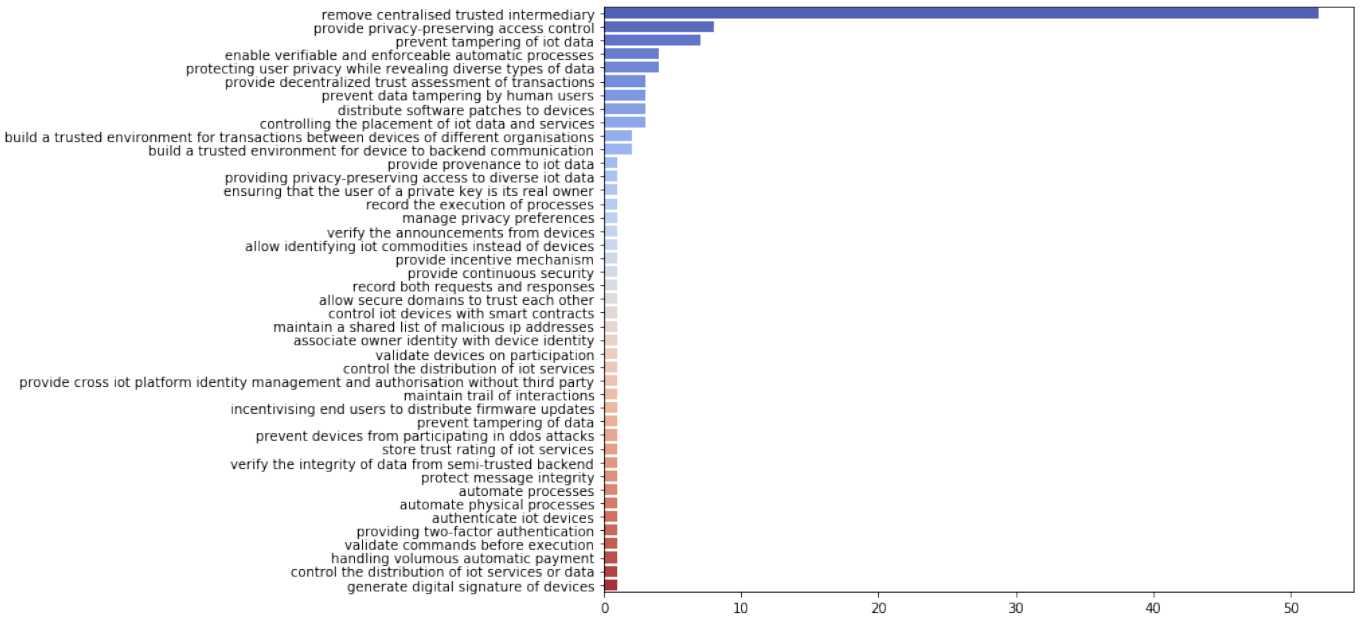

Fig. 5. Distribution of technical issues addressed by BC integration.