Architecting IoT systems supported by Blockchain

by

Wendy Yánez Pazmiño

A thesis submitted to
the University of Birmingham
for the degree of
DOCTOR OF PHILOSOPHY

School of Computer Science
College of Engineering and Physical Sciences
University of Birmingham
December 2021

ABSTRACT

Leveraging blockchain technology, IoT data can be recorded as immutable transactions and processed in consensus by blockchain nodes. Blockchain can ensure distributed and secure IoT data management due to its inherent features, such as transparency, auditability, traceability, and accountability. However, the implementation of blockchain in IoT systems is still facing some challenges. First, IoT systems are data-driven, characterized by high velocity, high volume of data, and high mobility, making data security an issue. Next, blockchain presents technical constraints of a complex nature, such as limited space, immutability, and excessive computational power, that can limit its adoption in IoT systems at scale.

Therefore, to address these challenges, a comprehensive investigation of architectural knowledge, design decisions, architectural tactics, styles, and data allocation mechanism that can drive the architectural design of IoT systems supported by blockchain is required. In this work, we identify the common quality attribute requirements, design decisions, and tradeoffs and their impact on system goals. We also present a catalog of architectural tactics that can help architects in achieving the quality attribute requirements of the system. In addition, we codify a set of reference architecture styles and variants for IoT systems supported by blockchain. Using a case study of healthcare, we evaluate the general fitness of styles with respect to quality attribute requirements using the Architecture Tradeoff Analysis Method (ATAM) and simulation. Finally, we propose a data allocation mechanism to dynamically decide on on-chain and off-chain data storage. The significance of this study is that it informs architects and designers with guidelines and blueprints on the architectural design of this category of systems by introducing a systematic investigation and evaluation approach.

To my family for being the strong pillar of my life, especially my mother for her endless love, support, and encouragement.

ACKNOWLEDGMENTS

First, I would like to thank my supervisor, Dr. Rami Bahsoon, who with his enthusiasm and experience conducting research supports my Ph.D. work. His continuous guidance, support, numerous ideas, sound advice, and patience have been invaluable to my research and have made this long journey an opportunity not only to learn at each meeting but also to get to know an extraordinary human being. I also thank my supervisor at SUSTech, Dr. Yuqun Zhang, for his support during my days on the other side of the world and kindly advice on my research work.

Special thanks are given to the members of my thesis group, Prof. Ela Claridge and Dr. David Galindo, for their continuous feedback and insightful comments on my research. I also thank Dr. Rajkumar Buyya, Prof. Rick Kazman, and Dr. Redowan Mahmud for their valuable time, fruitful discussion, and constructive feedback on the work we co-authored. I have had the privilege of sharing this journey with the wonderful people here at UoB. In particular, I would like to thank Dr. Sara Hassan for the long talks and valuable feedback; Dr. Dalia Sobhi, Dr. Rujia Li, and Alaa Alharbi for their timely comments and suggestions.

I am indebted to my parents, Merid and Augusto, for their unconditional support, love, and guidance; my sisters and brother, María José, María Sol, and Augusto Alexander, for their patience and inspiration; and my niece Sofía, for bringing happiness and memories to my life.

Finally, but not least important, a special thanks goes to Darth Edu for joining me on this journey and always believing in me!

Contents

List of Figures

List of Tables

Chapter One

Introduction

1.1 Overview

The Internet of Things (IoT) encompasses a network of physical objects known as "things" consisting of sensors, software, and network protocols to collect and exchange data with others and systems over the Internet [49]. These devices can range from simple sensors to sophisticated smartphones, wearables, kitchen appliances, and cars that enable a digital world with minimal human intervention. IoT is expected to connect 25 billion devices by 2020 and reach 100 billion by 2050 [78]. The rapid growth of IoT devices and recent advances in different technologies (i.e., cloud computing, big data, analytics and Artificial Intelligence -AI) have contributed to the development of IoT applications in a variety of domains such as smart cities, telehealth, manufacturing, and others [110].

However, the majority of IoT devices are vulnerable to a vast number of security issues due to their poor security-aware design, limited memory, battery lifetime and computational resources, and lack of standardization [89]. These issues have made it difficult for the IoT community to agree on a unique IoT reference model [40]. For example, Al-Fuqaha et al. [40] propose a five-layered IoT architecture that comprises the perception, object abstraction,

middleware, application, and business layer. Similarly, Qiu et al. [108] present a four-layer IoT architecture that includes the sensing, networking, cloud, and application layer. Unlike previous studies, Khari et al. [66] introduce a three-layer IoT architecture that includes the sensor, network, and application layer.

Although most of these studies rely on the cloud server as an independent or middleware layer for data processing and analysis, the cloud is also vulnerable to security and privacy issues, including data manipulation and unauthorized data sharing [106]. For example, a Facebook user's personal data leak occurred in 2018 due to cloud vulnerabilities [18]. Furthermore, sensitive information is only protected during transmission to the cloud and only a small part is encrypted [71]. In general, these studies indicate that current security issues in the IoT can also be associated with a lack of transparency and trust in cloud-based platforms. Therefore, it is critical for the future of IoT systems to move from a centralized architecture to a decentralized model to enable a trusted environment and ensure the integrity of your data.

In this regard, the blockchain is considered a potential technology to solve security issues in IoT systems. Blockchain offers a distributed ledger to record transactions and track assets in a Peer-to-Peer (P2P) network of computers called blockchain nodes [147]. An asset can be a physical (e.g., house, car, etc.) or non-physical (e.g., patents, intellectual property) object that can be monitored and traded on the blockchain network, reducing the risk of tampering [151]. Each blockchain consists of a chain of blocks, and each block includes data, its nonce, and hash, but also the hash of the previous block in the chain. The hash prevents any block from being altered and improves the verification of the previous block [89]. If a transaction is modified in one of the blocks, the blockchain nodes will reject it and the blockchain state will remain unchanged [109].

When a transaction is sent to the blockchain, it is represented in a block and trans-

mitted to the blockchain nodes, which validate the transaction using a consensus mechanism, including Proof-of-Work (PoW), Proof-of-Stake (PoS), Practical Byzantine Fault Tolerance (PBFT), etc. Once the transaction is confirmed to be legitimate, the block is added at the end of the existing chain and the update is distributed across the network [151]. It is not worth it that a blockchain network can include different types of nodes depending on their capabilities and resources, namely light, full and miner nodes [15]. *Light nodes* can only send and receive transactions, while *full nodes* can also store the complete copy of the blockchain. The full nodes can have additional capabilities to work as *miners* to create and mine blocks in the chain through a process called mining. All miners and full nodes on the blockchain network keep a copy of the entire ledger; therefore, trust is distributed among them.

Due to the inherent features of blockchain, such as immutability, transparency, integrity, and accountability, blockchain can guarantee the security of a variety of industries such as healthcare, supply chain, manufacturing, and others [78]. In particular, IoT data and message exchanges between heterogeneous IoT devices can be recorded as tamper-proof transactions and processed by a group of blockchain nodes in consensus [147]. The blockchain can also be used to set policies, monitor access rights to sensor data, and execute actions based on predefined conditions using smart contracts. [89]. Despite the above benefits, the deployment of blockchain technology in IoT systems still faces many challenges. First, most IoT devices have low memory, computation, and battery life, which means that they cannot directly implement blockchain functionality. Next, IoT systems are data-driven in nature, characterized by high velocity, high volume of data, and high mobility, making data security and its management a challenge. Furthermore, blockchain presents technical constraints of a complex nature that can limit its adoption in IoT systems on a scale, such as limited space, immutability, and excessive computational power, among others [89]. For example, Ethereum has a block size of around 2 MB on average and can handle only on average 3-20 transactions per second [21, 151]. These design issues that arise from the integration

of IoT and blockchain technologies require intensive analysis from a software architecture perspective.

Many research studies have analyzed the application of blockchain in IoT systems, such as healthcare [146, 76], smart transportation [157], smart vehicles [30], smart city [123], smart manufacturing [56], smart home [28], energy supply [74], and others, but only a few of them have considered the integration of these two technologies from the perspective of software architecture. For example, Xu et al. [151] conduct a systematic investigation on the architectural design issues of blockchain-based systems. In another work, Liao et al. [79] perform a comprehensive review of design issues to consider for the development of this category of systems. Reyna et al. [111] present a set of architectural alternatives for the integration of blockchain and IoT. Furthermore, Xiong et al. [147] suggest two architectural approaches to the integration of blockchain and IoT. In general, these studies highlight the need for a disciplined understanding of quality attributes, architectural tradeoffs, and design decisions that can drive the architectural design of IoT systems supported by blockchain.

To address these issues, firstly, this thesis reports on a systematic investigation of common quality attributes, architectural tradeoffs, and key design decisions to consider when designing IoT systems supported by blockchain. This comprehensive architectural knowledge can serve as a primary driver in the architectural design process for this category of systems. Second, the thesis investigates the architectural tactics that can help architects meet the desired quality attributes of interest when designing IoT systems supported by blockchain. The goal of the tactics is to provide a set of reusable architectures for software architects and developers to satisfy the desired qualities of the system. Third, the thesis codifies a set of reference architecture styles and variants for on- and off-chaining data and uses the Architecture Tradeoff Analysis Method (ATAM) to analyze the architectures in light of the desired system qualities and tradeoffs. Understanding common tradeoffs can assist software architects and designers in the choice of architectural styles to underlie IoT systems supported

by blockchain and to provide pre-architectural evaluation for realizing quality attributes of interest and tradeoffs supporting the systems. The results of the ATAM have led to refinements of the styles. Moreover, we support ATAM analysis with simulation to overcome the limitations of not having stakeholders who state the quality attribute requirements and elicit scenarios. Finally, the thesis develops a novel data allocation mechanism that relies on fuzzy logic and context information to decide which data need to be recorded on the blockchain (i.e., on-chain) or in external storage (i.e., off-chain). We define *on-chain storage* as the ability to store information in the blockchain itself and *off-chain storage* as the use of private/public cloud, local storage or peer-to-peer storage to keep information out of the blockchain and verify it through the blockchain.

1.2 Problems Addressed

The architectural design of IoT systems supported by blockchain should consider quality requirements, design decisions, tradeoffs, and technical limitations of both technologies. The adoption of blockchain as the backbone architecture for IoT data management should be optimized to handle a large amount of data collected by heterogeneous devices while improving the security of sensitive data.

However, the deployment of blockchain in IoT systems requires addressing some issues. First, IoT networks, consisting of resource-constrained devices, can be vulnerable to cyberattacks, resulting in data theft, data forgery, and botnet attacks due to their limited computing, memory, and power resources, as well as poor-aware security design [89]. Next, IoT networks are subject to device mobility and network volatility, leading to data inconsistency, incompleteness, imprecision, and vagueness, which can negatively influence decision making [111]. Furthermore, IoT data can include sensitive information that can suffer from

data manipulation and unauthorized data sharing, since only 10% of the data are encrypted in the cloud [71]. Finally, blockchain has limited computational power and data storage space, making blockchain adoption difficult in IoT systems at scale [150].

Many surveys and some research studies have been published on the integration of blockchain in IoT systems [23, 25, 31, 53], but these works focus on general applications of blockchain in IoT. For example, Christidis and Devetsikiotis [23] propose the integration of blockchain and smart contracts for IoT systems. Similarly, Conoscenti et al. [25] conduct systematic literature on blockchain and its impact on IoT applications. In another work, Dorri et al. [31] present a blockchain-based architecture for a smart house to overcome Bitcoin problems. Therefore, to cover the gaps in the literature on the integration of IoT and blockchain, a comprehensive study is required on the architectural design of this category of systems.

We advocate that IoT systems can benefit from the distributed architecture of blockchain, as well as its inherent features such as immutability, integrity, transparency, and data accountability [147]. The blockchain can maintain an audit trail of the permission to access sensor data through smart contracts and track how IoT devices autonomously communicate with each other without the need for a centralized authority [89]. Furthermore, blockchain can be used for digital forensics in IoT applications such as supply chain or healthcare, where data collected by heterogeneous IoT devices must be verified to provide a transparent view of the investigation process, including the chain of custody [111].

Thus, this thesis reports on *a novel systematic investigation on the architectural knowledge, design decisions, architectural tactics, styles, and data allocation mechanism that can drive the architectural design of IoT systems supported by blockchain.*. In particular, the following aspects regarding the architectural design of IoT systems supported by blockchain require further investigation:

- **Problem 1:** The inadequacy of a disciplined understanding of the quality attribute requirements, tradeoffs, and design decisions that can drive the architectural design of IoT systems supported by blockchain.
- **Problem 2:** The general absence of a comprehensive investigation and body of architectural knowledge documenting the architectural tactics that can be used to build candidate architectures for IoT systems supported by blockchain that achieve particular quality attribute requirements.
- **Problem 3:** The general absence of systematic investigations on architectural styles and architectural evaluation approaches that can be used to understand the tradeoffs inherent in architectures, inform design refinements, and decide on architectural choices that effectively realize the desired quality attributes in the IoT system supported by blockchain.
- **Problem 4:** The need for dynamic data allocation mechanisms for IoT systems supported by blockchain to decide on on-chain and off-chain data allocations, considering context information, quality attributes, IoT constraints, and blockchain limitations.

1.3 Research Questions

In an effort to address the problems defined in Section 1.2, we formulate a set of research questions (RQ) as follows.

RQ1: What software quality attribute requirements, architectural tradeoffs, and design decisions are commonly discussed for the architectural design of IoT systems supported by blockchain?

RQ2: What architectural tactics can be documented from identified architectural

design decisions to build candidate architectures for IoT systems supported by blockchain that achieve particular quality attribute requirements?

RQ3: What reference architecture styles can be implied to guide the development of IoT systems supported by the blockchain? How can we assess the fitness of the reference architectures with respect to particular system qualities? What are the applications and usage domains that can benefit from the reference architecture styles?

RQ4: What are the design decisions driving the development of a dynamic data allocation mechanism for IoT systems supported by blockchain? How can a data allocation mechanism be effectively engineered in this category of systems, considering context information, quality attributes, IoT constraints, and inherent limitations of the blockchain? How can the reference architecture styles and variants be enriched with a data allocation mechanism to decide on on-chain and off-chain storage?

1.4 Research Methodology

This thesis follows the iterative process proposed in Design Science Research Methodology (DSRM) [101] to answer the research questions presented in Section 1.3. The main steps are discussed in the following.

- **Identifying the problem:** The first step is to acquire knowledge of IoT systems supported by blockchain. To this end, we conducted a systematic literature review (SLR) that covers the state of the art of IoT and blockchain and practical applications on the integration of both technologies from the perspective of software architecture. The review provides a holistic view of the research in progress and identifies room for improvement in the current literature that can drive the development of IoT systems

supported by blockchain. This problem has been formulated as a set of sub-problems described in Section 1.2.

- **Defining the objectives:** The main objective of the thesis is to *conduct a systematic investigation on the architectural knowledge, design decisions, architectural tactics, styles, and data allocation mechanism that can drive the architectural design of IoT systems supported by blockchain.* This objective has been formulated as a set of research questions described in Section 1.3. By achieving this goal, we will be able to provide systematic guidelines and documented artifacts to assist software architects and developers in designing IoT systems supported by blockchain to meet the goals of the system.

- **Designing and developing the contributions:** First, the thesis reports on a systematic investigation of common quality attributes, architectural tradeoffs, and key design decisions for IoT systems supported by blockchain. Furthermore, the thesis investigates the architectural tactics that can be derived from the design decisions identified in the SLR to promote particular qualities of interest. Moreover, the thesis partially uses the Architecture Tradeoff Analysis Method (ATAM) [65] to understand and document the design tradeoffs in the reference architecture styles. In particular, we perform a qualitative evaluation using ATAM with simulation to obtain strong evidence of the applicability of the styles. Finally, the thesis uses fuzzy logic [90, 8] and context awareness [99, 1] to formulate a data allocation mechanism to decide on on-chain and off-chain storage in light of context information, quality attributes of interest, and the constraints of IoT and blockchain technologies.

- **Demonstrating and evaluating the contributions:** To demonstrate the effectiveness of the contributions, we describe them along with their evaluation as follows:

  - *Catalog of tactics:* We conduct a systematic investigation of architectural design

decisions that can guide the development of IoT systems supported by blockchain. It includes the analysis of 100 primary studies identified in the SLR to review quality attributes, tradeoffs, and design decisions. Next, we codify the design decisions of the reported studies into architectural tactics to satisfy particular quality attributes, such as security, scalability, performance, and interoperability. Finally, we use the design science-based evaluation approach to reflect on the catalog of architectural tactics. This contribution addresses RQ1 and RQ2.

- *Reference architecture styles:* We provide a set of reference architecture styles and variants by inspecting representative examples from the literature that can serve as guidelines for the development of IoT systems supported by blockchain. In addition, we use ATAM to assess the general fitness of the styles in relation to the qualities of the system and inform the inception of some variants derived from them. Finally, we complement the ATAM evaluation with a simulation using FogBus [138] to obtain strong evidence of the suitability of styles and their variants. This contribution addresses RQ3.
- *Data allocation mechanism:* We present a data allocation mechanism that relies on context information and fuzzy logic to decide on which data should be recorded on the blockchain (i.e., on-chain) or external storage (i.e., off-chain). To demonstrate the feasibility of the approach, we perform a quantitative evaluation using FogBus [138] to compare the performance of the proposed data allocation mechanism against classical data management approaches in IoT systems supported by blockchain. This contribution addresses RQ4.

1.5 Contributions

This thesis makes some novel contributions to provide systematic guidelines and documented artifacts to assist software architects and developers in the architectural design of IoT systems supported by blockchain that achieve the desired system qualities. In particular, this thesis investigates commonly reported quality attribute requirements, design issues, architectural tactics, and a data allocation mechanism to facilitate the development of this category of systems. In summary, the main contributions of this thesis are as follows.

- *Architecting IoT systems supported by blockchain: A catalog of architectural tactics* – We investigate the common quality attributes, tradeoffs, and design decisions in the primary studies identified in the SLR of IoT systems supported by blockchain. The SLR results show security, performance, scalability, and interoperability as quality attributes that are commonly discussed in the identified studies. Similarly, we categorize design decisions as data and computation distribution, blockchain scope, consensus protocol, data structure, and blockchain deployment. These design decisions have led us to identify a set of architectural tactics to guide software architects and developers in the architectural design of IoT systems supported by blockchain. Finally, we use the design science-based evaluation approach to reflect on the catalog of architectural tactics.

- *Reference architecture styles for IoT systems supported by blockchain* — We codify a set of reference architecture styles and variants for on- and off-chain data. Moreover, we use ATAM analysis to inform the inception of some variants derived from reference styles to satisfy particular qualities of the system. In particular, the ATAM results have explicitly identified tradeoff points among quality attribute requirements in the styles and documented them as refinements of the architectures. We support ATAM analysis with simulation to overcome the limitations of not having stakeholders that state the quality attribute requirements and elicit scenarios. In general, the identified

reference architectures and variants can be of great value to software architects and developers in reasoning about design decisions and quality attributes and guide the design of this category of systems.

- *Data allocation mechanism for IoT systems supported by blockchain* — We develop a data allocation mechanism that deals with IoT constraints (i.e., high mobility, high velocity, and high data volume) and inherent limitations of the blockchain (e.g., limited computational power and data storage). The mechanism implements a controller that uses context information and fuzzy logic to decide which data needs to be recorded on the blockchain (i.e., on-chain) or in external storage (i.e., off-chain). In particular, the controller extracts context information from IoT data (i.e., data, network, and quality) and uses fuzzy to calculate the Rating of Allocation (RoA) value, which serves as a threshold to make allocation decisions.

1.6 Publications

This thesis compiles research work that has been previously published or is currently being submitted to highly competitive journals. The following publications summarize the research ideas and developments of the thesis.

- Yánez, W., Bahsoon, R., Zhang, Y., & Kazman, R. (2021). Architecting Internet of Things Systems with Blockchain: A Catalog of Tactics. ACM Transactions on Software Engineering and Methodology, (TOSEM), 30(3), 1-46.

- Yánez, W., Mahmud, R., Bahsoon, R., Zhang, Y., & Buyya, R. (2020). Data allocation mechanism for Internet-of-Things systems with blockchain. IEEE Internet of Things Journal, 7(4), 3509-3522.

- Yánez, W., Li, R., Bahsoon, R., Zhang, Y., & Kazman, R. (2021). Architectural Styles for the Integration of Blockchain on the Internet of Things. IEEE Internet of Things Journal, (IEEE IoT Journal), (To be submitted).

1.7 Roadmap

Figure 1.1 shows how the research questions are related to the chapters of this thesis.

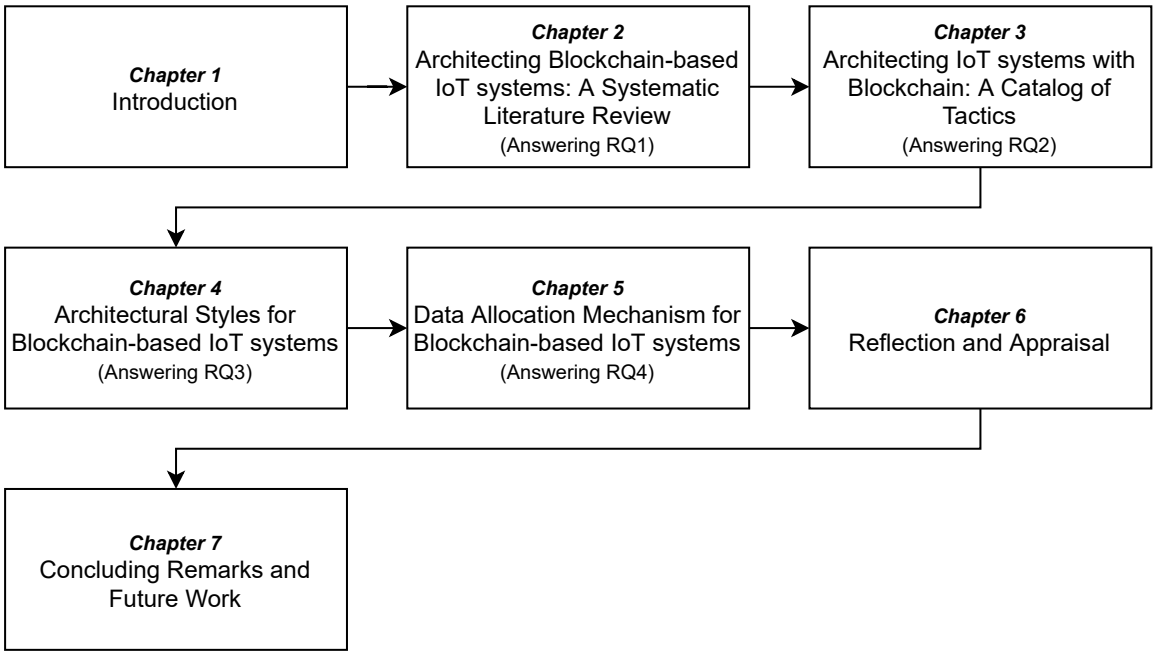

Figure 1.1: Thesis at a glance.

The remainder of the thesis is summarized below.

- *Chapter 2 - Architecting IoT systems supported by blockchain: A Systematic Literature Review.* This chapter presents the result of the Systematic Literature Review (SLR) that investigates the requirements, tradeoffs, and architectural design decisions reported in the literature. First, we present the objectives of the study and the research methodology, which follows the guidelines suggested by [70] and [102]. Next,

we analyse the primary studies and identify security, scalability, and performance as the commonly discussed quality attribute requirements that have driven the development of IoT systems supported by blockchain. We also categorised architectural design decisions to achieve the quality attributes of the system, expressed as data and computation distribution, blockchain scope, consensus protocol, data structure, and blockchain deployment. Finally, we infer research gaps in the current architectural design of this category of systems and address them in the rest of this thesis. This chapter is partially derived from [152].

- *Chapter 3 - Architecting IoT systems supported by the blockchain: A Catalog of Tactics.* This chapter presents a catalog of architectural tactics derived from the design decisions identified in the primary studies. In general, designers and architects can reuse tactics to satisfy particular quality attributes in IoT systems supported by blockchain. First, we categorize architectural tactics in light of the qualities of the system (i.e., security, performance, and interoperability). Next, we describe the identified tactics using the template proposed by Lewis and Lago [73], which includes a brief description, constraints, examples, related tactic, and variations (optional). Additionally, we use the design science-based evaluation approach to reflect on the catalog of architectural tactics. Finally, we identify gaps and opportunities for research, including evaluation of the real-world impact of the identified architectural tactics in the existing architectures and exploration of the tradeoffs among the quality attributes and identified tactics. This chapter is derived in part from the work presented in [152].

- *Chapter 4 - Reference architecture styles for IoT systems supported by blockchain.* This chapter presents a set of architectural styles and variants by inspecting representative examples from the literature to facilitate the development of IoT systems supported by blockchain. First, we investigate the state-of-the-art and applications on the integration of blockchain and IoT to extract the underlying architectural styles that support

them, as well as the design decisions that lead to the styles. Next, we use ATAM to guide the inception of variants, informed by systematic analysis of tradeoffs between quality attributes promoted by this category of systems that needs to be achieved. The ATAM is a scenario-based method for evaluating candidate architectures relative to quality attributes requirements. We complement the ATAM analysis with simulation to assess the general fitness of the styles and their variants in terms of quality attributes, concerted as scenarios, and their tradeoffs. Finally, we discuss the results of the ATAM evaluation and simulation, as well as infer gaps in the architectural development of IoT systems supported by blockchain. This chapter is partially derived from the work presented in [154].

- *Chapter 5 - Data allocation mechanism for IoT systems supported by blockchain.* This chapter presents a data allocation mechanism to decide on which data should be recorded on the blockchain (i.e., on-chain) or external storage (i.e., off-chain). The mechanism combines context information and fuzzy logic to make strategic allocation decisions. In particular, it implements a fuzzy logic-based controller that extracts context information from IoT data (i.e., data, network, and quality) to decide on its allocation. Then, we introduce the mechanism in two existing architectural styles and compare their performance in terms of blockchain size, latency, energy consumption, and network usage. Both the high-level description and the evaluation of the mechanism are presented, along with the refinements of the styles. This chapter is derived from the work presented in [153].

- *Chapter 6 - Reflection and Appraisal.* This chapter systematically evaluates the general thesis by describing how the research questions in Section 1.3 were addressed. We also discuss the architectural aspects of the simulation developed as part of the research to demonstrate the effectiveness and suitability of the proposed approaches.

- *Chapter 7 - Concluding Remarks and Future Work.* This chapter concludes the thesis

with a summary of the main contributions and a brief discussion of the main findings and observations related to the proposed research questions. We also present an outlook for future directions in this domain.

Chapter Two

Architecting IoT systems supported by blockchain: A systematic review of the literature

This chapter describes the protocol and results of a Systematic Literature Review (SLR) to investigate the most relevant architectural design decisions in IoT systems supported by blockchain. It includes a comprehensible analysis of 100 primary studies to identify common quality attributes, tradeoffs, and design decisions that can guide the development of this category of systems. In particular, we categorize design decisions as data and computation distribution, blockchain scope, consensus protocol, data structure, and blockchain deployment. The results show some gaps and opportunities for research that we expect to address in the remainder of this thesis.

2.1 Overview

Architectural design is the process of helping architects make strategic design decisions. It involves reasoning about a system to understand the main software elements and the relationships among them [126]. This process is particularly relevant in complex systems, such as IoT systems supported by blockchain, in which IoT is characterized by some constraints (i.e., mobility, high velocity, and large data volume), and blockchain presents some inherent limitations (e.g., limited computational power and data storage) [147]. Hence, understanding the system qualities, tradeoffs, design decisions, the rationale behind these decisions, and the context in which they are conceived is key for moving towards the architectural design of a software system.

Several attempts have been made to use blockchain in IoT systems [146, 157, 30, 123, 56, 28, 74]; however, only a few analyze design issues for architecting IoT systems supported by blockchain. Among these works, Xu et al. [151] conduct a systematic investigation of the architectural design issues of blockchain-based systems. In another work, Liao et al. [79] conduct a comprehensive review of design issues to consider for the development of this category of systems. Reyna et al. [111] present a set of architectural alternatives for the integration of blockchain and IoT. Furthermore, Xiong et al. [147] review two architectural approaches to the integration of blockchain and IoT. In general, these studies highlight the analysis of architectural design issues for blockchain-based systems and a set of architectural alternatives to support the integration of blockchain and IoT. However, they have tended to focus on systematic investigation of design decisions and their impact on quality attributes rather than on the implementation and evaluation of candidate architectures. Therefore, we conclude that there is still a lack of a disciplined understanding of the quality attributes, architectural tradeoffs, and design decisions that can drive the architectural design of IoT systems supported by blockchain.

This chapter reports on a systematic investigation of the commonly identified quality attributes, architectural tradeoffs, and key design decisions for IoT systems supported by blockchain. It also contributes to a comprehensive and novel catalog of architectural knowledge to guide the development of this category of systems. In particular, we conducted an SRL to investigate common software quality attributes, architectural tradeoffs, and design decisions discussed in primary studies that can guide the development of IoT systems supported by blockchain. The SLR follows the guidelines proposed in [70] and [102] to identify the existing work on design decisions and how they affect the quality attributes of IoT systems supported by blockchain. In addition, we state the inclusion and exclusion criteria before starting the analysis to assess the research. Our findings are drawn from 100 research publications that are rigorously selected from a repository of 575 peer-reviewed, published articles on blockchain and IoT. The results of the SLR show quality attributes and design decisions reported across IoT systems supported by blockchain, as well as opportunities for research in the domain.

We identify security, scalability, performance, and interoperability as the commonly discussed quality attributes in IoT systems supported by blockchain. Additionally, we categorize design decisions such as data storage and computation distribution, blockchain scope, consensus protocol, and blockchain implementation. The review also allows us to identify gaps and opportunities for research regarding (i) the lack of architectural support for some quality attributes (e.g., mobility, interoperability, and others), (ii) identification of architectural tactics and styles that support IoT systems supported by blockchain, (iii) limited evaluation of the impact of the architectural tactics in this category of systems, and (iv) inadequacies in the analysis of tradeoff points among identified quality attributes and tactics. This chapter presents the following contributions.

- A set of common quality attribute requirements reported in the literature that are

relevant for the development of IoT systems supported by blockchain.

- A set of architectural design decisions identified in primary studies that can guide the design of this category of systems. Decisions were classified as data distribution and computation, consensus protocol, blockchain scope, and blockchain deployment.

- A set of gaps and opportunities for research that we expect to address in the remainder of the thesis, including the identification of architectural tactics and styles that support IoT systems supported by blockchain and inadequacies in the analysis of tradeoffs points among identified quality attributes and tactics.

The remainder of this chapter is organized as follows. Sections 2.2 and 2.3 describe the research method and the analysis of the primary studies, respectively. Section 2.4 discusses the main findings and potential areas for future research. Sections 2.5 and 2.6 summarize the threats to the validity of previous efforts on the design of IoT systems supported by blockchain, respectively. Finally, Section 2.7 concludes the chapter.

2.2 Research Protocol

We conducted an SLR of the common software quality attributes and tradeoffs between them in IoT systems supported by blockchain reported in the literature. Bass et al. [12] have defined a quality attribute as "a measurable property of a system to evaluate how well it satisfies business objectives". In particular, we used this definition to argue about the quality attribute requirements and their tradeoffs necessary for deployment of this category of systems. To develop our review protocol, we follow the guidelines and procedures suggested by Kitchenham et al. [70] and Petersen et al. [102], as well as the work of [73]. In summary, the protocol included (i) research questions, (ii) search strategy, (iii) inclusion and

exclusion criteria, (iv) study selection, and (v) data extraction and synthesis procedures. These systematic guidelines allowed us to conduct a systematic exploration of the current body of knowledge and select representative studies in the field. Furthermore, it is important to highlight that the review protocol was developed by one of the coauthors and was revised by others to limit bias.

2.2.1 Research Questions

The goal of the SLR is to identify the common quality attributes (RQ1) and design decisions (RQ2) necessary to design IoT systems supported by blockchain that have been reported in the literature.

- *RQ1: What are the most common quality attributes and tradeoffs that must be considered when designing IoT systems supported by blockchain?*

  Aim: Identify the commonly quality attribute requirements and the tradeoff between them that have been reported in the literature and must be met to design IoT systems supported by blockchain.

  Relevance: By answering this question, we can help architects and designers to reason about (i) the quality attributes necessary to design this category of systems and (ii) an overview of the possible tradeoffs among them.

- *RQ2: What are the relevant architectural design decisions that influence the achievement of quality attribute requirements in IoT systems supported by the blockchain?*

  Aim: Investigate the design decisions that are considered in the development of this category of systems to achieve the desired quality attributes.

Relevance: The result of this research question can provide architects and designers with a clear understanding of the architectural design decisions to be made to achieve quality attributes.

2.2.2 Search Strategy

We formulate a general search string derived from the proposed research questions, which includes the following terms and closely related alternative terms such as (i) *IoT*, (ii) *blockchain* and (iii) *software architecture*. We also considered some additional keywords, such as fog computing and edge computing, to obtain information on potential computing infrastructures where blockchain networks can be deployed and implemented. Finally, we combined these terms interchangeably and created the following search string, which was tested against a set of known primary studies to assess its reliability.

(internet of things OR internet of thing OR iot) AND (blockchain OR blockchain technology OR distributed ledger technology OR DLT) AND (Fog computing OR fog OR edge computing OR edge) AND (software architecture OR software design OR software requirements OR architectural styles OR patterns OR reference architectures)

We carried out our automatic search on electronic databases and indexing libraries (i.e. IEEE Explore, ACM Digital Library, Scopus, and Web of Science). These databases were selected based on (i) the variety of electronic resources and library catalogs they provide to support software engineering research [102, 69] and (ii) their popularity among systematic mapping studies in software engineering [102]. Our work focused on high-quality and advanced journals, conference proceedings, and scientific workshops, while excluding other sources such as books, thesis, talks, blogs, and presentations that provide irrelevant information. Furthermore, our preliminary search was not restricted to the publication date to

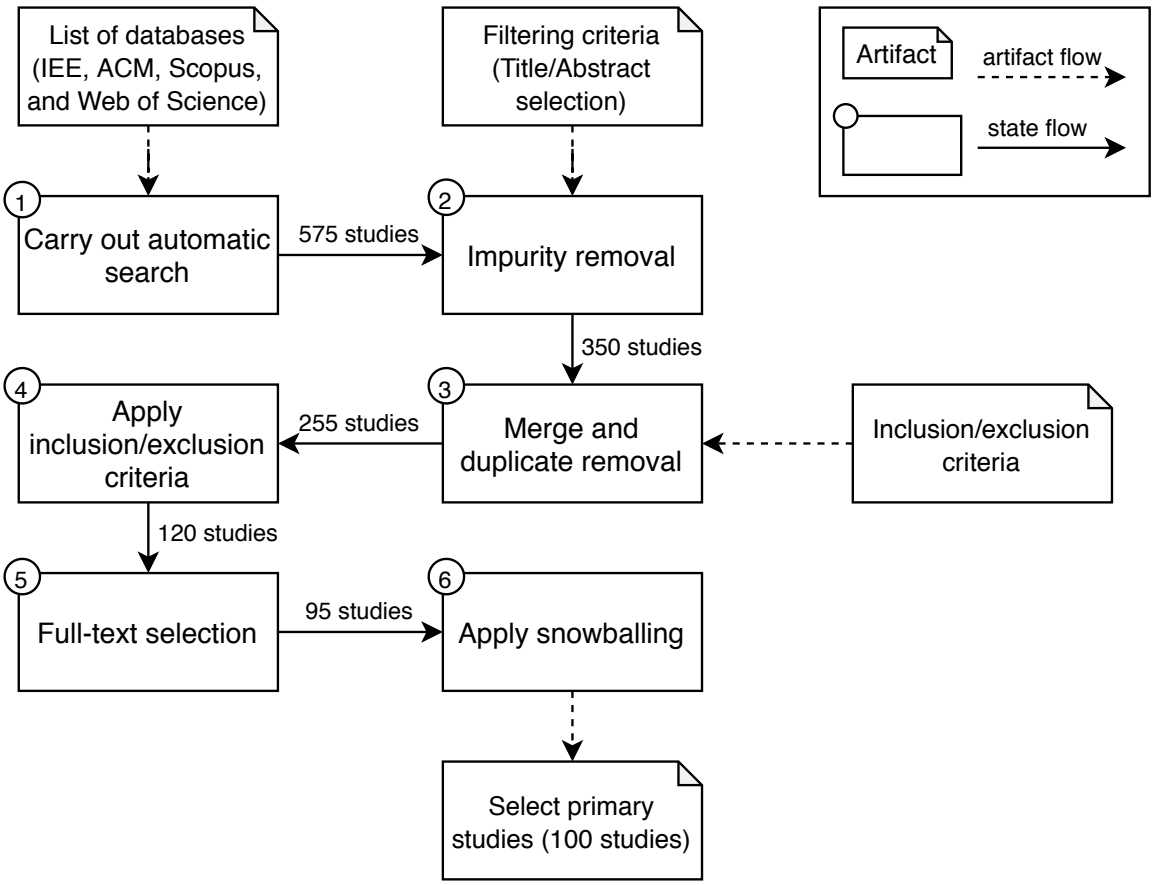

Figure 2.1: Overview of the selection process.

allow for a broad coverage of studies related to research questions of interest. We use the Publish or Perish [51] tool to retrieve academic results from selected digital libraries and maintain metadata for further analysis.

2.2.3 Study selection

We defined a selection procedure to identify studies that provide direct evidence of the proposed research questions. This procedure was discussed and revised by the supervisor, members of the software engineering team, and experts in the field. Figure 2.1 shows the number of studies included and excluded in the selection process.

1. *Initial search:* We retrieved a total of 575 relevant studies from the IEEE Explore, ACM Digital Library, Scopus, and Web of Science databases using the search string above.

2. *Impurity removal:* We manually removed studies that were not relevant to the proposed research questions based on the title and abstract. This process was carried out by the first and second authors to decide the studies for the next round, resulting in 350 of the 575 initial studies.

3. *Merging and duplication removal:* The selected studies were analyzed by the first and second authors to remove duplicates because some publications were also available in IEEE Explorer and ACM. As a result, we created a single dataset of 255 studies to be used in the next round of our selection process.

4. *Selection criteria:* Table 2.1 describes the inclusion and exclusion criteria that the first and second authors applied to all selected studies, resulting in a total of 120 studies.

5. *Full-text selection:* The first and second authors read the full text of the selected studies to ensure their alignment with the research questions and eliminate bias. As a result, the number of candidate studies was reduced to 95 and shared with the software engineering group for evaluation.

6. *Snowballing:* We complemented our full text reading with recursive backward and forward snowball activities described by Wohlin [142] to complement the automatic search. In the backward snowballing, we focused on the references of the primary studies, while in the forward snowballing, we used Google Scholar to obtain new publication results. As a result, a total of five studies were added to the final set and evaluated using inclusion and exclusion criteria.

Although the exclusion criteria E2 in Table 2.1 remove secondary studies, we still

consider them to (i) identify their contribution to the study, (ii) define as many studies related to research questions, and (iii) investigate their relevance to the architectural design of IoT systems supported by blockchain.

Table 2.1: Inclusion and exclusion criteria.

#	**Inclusion criteria**
I1	Discuss quality attribute requirements necessary for architecting IoT systems supported by blockchain.
I2	Provide software architecture solutions, including tactics and styles for the design of this category of systems.
I3	Present evaluation of the architectural strategies, methods, or techniques for the architectural design of this category of systems (e.g., case scenarios, prototype solutions, simulations, etc.).
I4	Subject to peer-review.
I5	Written in English.
#	**Exclusion criteria**
E1	Propose the integration of blockchain and IoT, but do not present an architecture.
E2	Include blockchain and IoT as secondary studies (e.g., systematic literature review, surveys, etc.).
E3	Present as tutorial papers and editorials that were not in the form of a published paper, which do not provide direct evidence of the integration of blockchain and IoT.
E4	Full papers that document the approach and provide potential evaluation.

2.2.4 Data extraction

We designed a framework to rigorously extract information on the research questions proposed from primary studies. In particular, the first author performed the data extraction procedure, while the second author selected a random sample of primary studies to cross-check the results with those of the first author. This process reduces threads to reliability and bias in the selection procedure (see Section 2.5). Table 2.2 describes the data extraction fields with their corresponding values and related research questions.

Table 2.2: Data extraction form.

Data item	**Value**	**RQ**
Study ID	Number	
Study title	Name of the study	
Author name	Author(s) in the study	
Publication year	Number	
Publication type	Conference, journal and workshop	
quality attribute requirements	Quality attributes identified in the studies.	RQ1
Architectural design	Design decisions towards the integration of blockchain and IoT.	RQ2

2.3 Analysis of the primary studies

This section presents the results of primary studies to identify the quality attributes, trade-offs, and design decisions that are necessary to design IoT systems supported by blockchain.

2.3.1 Quality attributes for architecting IoT systems supported by blockchain

Table 2.3 presents the commonly reported quality attributes and tradeoffs between them to consider when designing IoT systems supported by blockchain and some examples (for detailed explanations, see Section 2.2). It should be noted that not all studies have explicitly mentioned the quality attributes that they address to realize the functionality of the system. In most cases, we identify them by looking at the primary studies in detail and relating their solution to the user and system requirements. Furthermore, some studies focus on achieving more than one quality attribute. For example, [150] considers performance and security as the most important quality attribute requirements to satisfy in this category of systems, while highlighting security as a critical quality requirement [30]. In the reviewed literature, we identify security, scalability, and performance as commonly reported quality attributes with a total of 55, 23, and 18 studies, respectively (see Table 2.3). In addition to these quality attributes, there are others that appear in some studies, such as interoperability, mobility, adaptability, and efficiency, which could also have a significant impact on the architectural design of IoT systems supported by blockchain. For each quality attribute, we provide a brief explanation and a reason for its importance in the architectural design of IoT systems supported by blockchain, as follows.

Security

According to Barbacci et al. [11], security mainly comprises three concerns: confidentiality, integrity, and availability. Confidentiality refers to protecting data from unauthorized disclosure, while integrity prevents unauthorized data modification. Similarly, availability ensures data access to authorized users. Therefore, an ideal IoT system supported by blockchain must implement access control permissions via smart contracts to restrict access to only

Table 2.3: Quality attributes.

Quality attributes	**% of studies**	**Representative examples**
Security	55%	[PS1, PS3, PS5, PS6, PS7, PS10, PS11, PS12, PS15, PS18, PS19, PS21, PS23, PS24, PS25, PS27, PS28, PS29, PS30, PS32, PS33, PS34, PS35, PS36, PS39, PS40, PS41, PS45, PS47, PS48, PS49, PS51, PS52, PS55, PS57, PS58, PS62, PS65, PS66, PS69, PS70, PS72, PS75, PS77, PS78, PS80, PS81, PS85, PS86, PS88, PS90, PS91, PS93, PS95, PS99]
Scalability	23%	[PS5, PS7, PS11, PS12, PS15, PS19, PS26, PS27, PS29, PS30, PS32, PS34, PS36, PS38, PS40, PS42, PS43, PS48, PS49, PS52, PS60, PS61, PS6]
Performance	18%	[PS2, PS23, PS25, PS37, PS38, PS41, PS50, PS56, PS57, PS63, PS67, PS68, PS76, PS79, PS82, PS86, PS92, PS93]

authorized participants (confidentiality) and keep critical data and raw data hashes on the blockchain to ensure its immutability and integrity (integrity). Furthermore, this category of systems must replicate sensor data across the P2P network to ensure its availability to authorized participants (availability) [89, 111].

Scalability (concerning blockchain size and transaction throughput)

An optimal IoT system supported by blockchain must ideally achieve low transaction throughput with the increase in the number of miners and validator nodes in the blockchain network. However, increasing the number of blockchain nodes could increase the number of transac-

tions, thus increasing the size of the blockchain. With increasing blockchain size, storage requirements also increase. It could create more limitations on the integration of resource-constrained IoT devices to act as miner nodes in the blockchain network. Furthermore, increasing the size of the blockchain could result in longer synchronization for new devices or users who want to join the blockchain network [89, 113].

Performance (concerning latency in transaction confirmation)

An ideal IoT system supported by blockchain must achieve low latency in transaction confirmation to ensure instant consensus agreement, which is a fundamental requirement in most real-time IoT systems, such as smart vehicles, smart grids, and intelligent transportation systems. A possible way to minimize the transaction confirmation time while achieving the same level of security is by reducing the block generation time. However, it could require waiting for more confirmations due to the lower difficulty of mining a block. The latency could also be reduced by increasing the block size. For example, on the Bitcoin blockchain, the block size can be increased from 1 to 2 MB to improve throughput in the network, but it will lead to longer blocks that could be difficult to propagate in the blockchain network. Furthermore, increasing block size will result in a continuous increase in the size of the blockchain, resulting in more full nodes with high storage capacity to store a copy of the complete blockchain [89].

Interoperability

An ideal IoT system supported by blockchain must ensure data exchange between different blockchain implementations and the integration of heterogeneous devices as blockchain nodes [4, 111]. Specifically, multiple blockchains can be used to allow the separation of concerns among different types of transactions and business goals, but their interaction must be guar-

anteed to meet the requirements of IoT systems [150]. Furthermore, IoT devices that work as full or lightweight blockchain nodes should be able to communicate and share information with nodes in another chain [122].

Efficiency

An optimal IoT system supported by blockchain must ensure a cost-effective and efficient utilization of hardware and power resources in IoT devices and blockchain nodes [37, 113]. On the one hand, reducing redundant data movements from IoT devices to the cloud could minimize latency and energy consumption in the system [111]. However, the selection of resource-intensive consensus protocols such as Proof-of-Work (PoW) could impose new challenges in the adoption of blockchain in IoT systems due to limited resources in most IoT devices. Thus, a lightweight consensus protocol and an alternative verification mechanism could be required that has a small footprint and low energy costs [31].

Adaptability

A data-centric IoT system supported by blockchain must adapt IoT networks and rules into smart contracts based on user and system requirements. Specifically, adaptability in IoT refers to dynamic traffic in IoT networks and heterogeneous features in IoT devices (i.e. different software and hardware resources) that allow them to join and leave the network [107]. It makes it easier for attackers to compromise IoT devices with fake IDs and manipulate IoT networks in the presence of such networks. Therefore, IoT networks must continuously adapt to changes in traffic load and uncertainties in environmental conditions. For blockchains, adaptability means changes in business logic (i.e. rules and agreements) on-chain stored in smart contracts based on the environmental context [84]. However, if the blockchain is used mainly as secure storage, then adaptability in smart contracts does not need to be ensured.

Mobility

An ideal IoT system supported by blockchain must be able to handle the mobile aspect of most sensors and IoT devices that change locations according to hardware resources and system requirements. Similarly, mobility in blockchain means having intermediate energy distributors, analytical or storage, to reduce the computation and storage loads on blockchain nodes and improve energy efficiency in this category of systems [89].

2.3.2 Categorization of the architectural decisions

An architectural decision should be accompanied by the rationale for the decision, expressed in terms of how this decision helps achieve one or more desired quality attributes, together with any drawbacks or tradeoffs [12]. Table 2.4 summarizes the main design decisions related to blockchain identified in the primary studies and their impact on the desired quality attribute requirements necessary for the development of IoT systems supported by blockchain [150].

Table 2.4: Design decisions for blockchain-based systems.

Design decision	**Quality attributes and tradeoffs**	**Impact**
Data storage and computation: What data and computation should be placed on-chain (i.e., within the blockchain and off-chain (i.e., external storage like cloud)	*On-chain:* Enhances security of IoT data, but it is computationally expensive and energy hungry.	Limit the amount of data that can be stored on-chain.
	Off-chain: Improves the scalability and availability of the blockchain, but represents a high maintenance cost and requires additional trust.	Interaction issues between on-chain and off-chain storage.
Blockchain scope: What type of blockchain should be used?	*Public:* Ensures data transparency and auditability, but potentially poor performance (i.e., high transaction confirmation cost and limited block size).	Privacy and confidentiality concerns since data is available to all blockchain nodes.

Continued on next page

Table 2.4 – *Continued from previous page*

Design decision	**Quality attribute and tradeoffs**	**Impact**
	Private: Improves the performance of the blockchain network, but offers little support for data auditability and transparency.	Centralization issues since the data is managed by a single entity.
	Consortium: Managed by multiple organizations and ensures better performance, scalability, and security.	Has the same advantages of a private blockchain but operates under the leadership of a group.
Consensus protocols: Which consensus protocol should be selected?	*Proof-of-Work (PoW):* Computationally expensive and time consuming.	Require powerful hardware for mining transactions.
	Proof-of-Stake (PoS): Improves performance and requires less computing power and energy, but extensive control and authority over technical and economic aspects by participants could lead to a monopoly problem.	Centralization of voting power results controlling the blockchain network.
	Practical Byzantine Fault Tolerance (PBFT): Improves security and performance, but affects scalability.	Single-point-of-failure due to the size of the blockchain network.

Continued on next page

Table 2.4 – *Continued from previous page*

Design decision	**Quality attribute and tradeoffs**	**Impact**
	Proof-of-Authority (PoA): Improves security because an authority is assigned a fixed time slot within which it can generate blocks.	Assume trusted authorities.
Blockchain data structure: Which type of data structure should be configured (e.g., single chain or multiple chain)	*Single chain:* Easy chain management and permission control, but it makes complex data management.	With the increasing number of transactions from IoT devices, a single blockchain might become overloaded and make data retrieval difficult.
	Multiple chains: Easy data management, but makes chain management and permission control harder.	Allows recording of IoT data in different blockchains for easy data storage and retrieval.
Blockchain deployment: Where should the blockchain be deployed?	*IoT:* Improve the scalability of the blockchain, but this leads to performance issues.	Enable IoT devices to work as nodes of the blockchain network.
	Fog: Improves the scalability and performance of the blockchain network, but leads to management problems.	Ensures decentralization in the end-to-end system.

Continued on next page

Table 2.4 – *Continued from previous page*

Design decision	**Quality attribute and tradeoffs**	**Impact**
	Cloud: Ensure decentralization and improve security in the cloud, but leads to high latency and bandwidth consumption.	Enable a large amount of computing resources.

2.3.3 Architectural decisions to consider in IoT systems supported by blockchain

The commonly reported architectural decisions for the design of IoT systems supported by blockchain, as defined earlier in Section 2.2 are summarized as follows:

Distribution of computation and storage

One of the major design decisions in the development of IoT systems supported by blockchain is what data and computation should be kept on-chain or recorded off-chain [79, 84]. In particular, the use of on-chain and off-chain storage should consider limited computation (transaction throughput) and data storage (block size) in public blockchains [150]. For example, Bitcoin has a block size of 1 MB and can only handle 7 transactions per second (TPS) on average, while VISA can perform 60,000 TPS on average. Furthermore, data are replicated on blockchain nodes, and the use of blockchain storage is expensive compared to

the use of conventional storage systems (i.e., local database, cloud, or P2P storage) [52]. Therefore, many studies in the literature store hashes of the raw data in the blockchain or record IoT data in smart contracts. Other studies use off-chain storage to record raw data generated by IoT devices rather than keeping the full data on the blockchain. Table 2.5 shows the distribution of the IoT data on-chain and off-chain, along with some representative studies in each category.

Table 2.5: Distribution of computation and storage.

Design decision	**Option**	**Representative example**
On-chain	Transactions and smart contracts	[PS1, PS5, PS7, PS12, PS13, PS16, PS22, PS25, PS26, PS27, PS29, PS30, PS31, PS32, PS34, PS35, PS37, PS38, PS39, PS40, PS42, PS45, PS47, PS49, PS51, PS52, PS53, PS55, PS56, PS60, PS62, PS68, PS71, PS72, PS74, PS75, PS77, PS78, PS79, PS81, PS82, PS88, PS91, PS93, PS95, PS97, PS98]
On-chain/Off-chain	Transactions and smart contracts/Cloud, local database, and P2P storage	[PS18, PS70, PS43, PS44, PS57, PS66, PS69, PS83, PS87]

On-chain. A common practice for data management in blockchain-based systems is to store small critical data, hashes of the raw data, and meta-data in the blockchain [84]. Data can be packed into (i) a transaction or (ii) a smart contract.

- **Recording data as transaction:** Due to the limited storage of data on the blockchain, a small amount of data can be stored on-chain or as part of a transaction, [84, 150]. These systems include Blockchain for data sharing [PS9], Blockchain Transportation [PS1], IoT updates [PS7], Optimized blockchain [PS2], and MeDShare [PS4]. MediChainTM [PS50] is a special case because it stores metadata and hashes of raw data on-chain to ensure its integrity and immutability. However, many of these systems do not explicitly mention which data is stored on-chain, which imposes new challenges in the development of future IoT systems supported by blockchain.

- **Recording IoT data via smart contracts:** A smart contract is a general program that can codify the states of physical assets or data exchanges between IoT devices [79]. However, the storage of a large amount of logic or data in the blockchain could lead to high transaction throughput in transaction processing, since most blockchain nodes need to reach a consensus to validate them. The following are examples of studies that use on-chain data storage through smart contracts, i.e., Blockchain Transportation [PS1], MIoT [PS39], Auth IoT [PS61], and MediChain [PS50]. However, the use of smart contracts on blockchains has a deployment and execution cost that must be considered when designing and architecting IoT systems supported by blockchain.

Off-chain. Due to limited data storage on public blockchains, raw data, data request source, smart contract addresses, and smart contract code are usually stored off-chain (i.e., local database, cloud or P2P storage) [121]. These storage solutions have their advantages and disadvantages in terms of transparency, storage cost, and centralization. The following are a set of studies that rely on cloud platforms as off-chain storage, for instance, Optimized blockchain [PS2], Blockchain for data sharing [PS9], Vegvisir [PS11], IoT data assurance [PS18]. MediChain [PS50] is a special case because it encrypts sensitive data (e.g., diagnostic images, lab test results, prescript, treatment plans) before storing it in a remote resource (i.e.,

enterprise cloud or data center), which are located in multi-hop proximity to IoT devices. Similarly, MeDShare [PS4], Forensic SDN [PS36], and Hybrid IoT [PS43] use a local database located in a single hop proximity to IoT devices as off-chain storage. The last set of systems uses P2P storage to record IoT data and includes auditable blockchain storage [PS8], IoT protection-blockchain [PS14], and Emergency SH [PS33].

Many studies combine the use of on- and off-chain storage as follows. In [PS8], a blockchain-based auditable storage and sharing of IoT data is presented to record access control permissions on the blockchain while keeping IoT data streams in off-chain storage. In another work [PS9], a blockchain-based data sharing and collaboration application is developed to protect privacy and allow identity management. In particular, healthcare data is stored in the cloud, and a proof of integrity is retrieved from the cloud and anchored to the blockchain network. Similarly, the authors in [PS18] propose a distributed solution based on blockchain to secure drone communication and data transmission. In particular, the hashes of the drone data are stored in the blockchain network while keeping the drone data and a receipt of each record in the cloud.

Blockchain configuration

This comprises a set of design decisions (i.e., type of blockchain, consensus protocol, and data structure) to consider when implementing blockchain-based systems. Table 2.6 summarizes the results of the design decisions of the blockchain configuration, with some representative examples under each category.

Table 2.6: Design decisions for blockchain-based systems.

Data structure	**Type of blockchain**	**Consensus Protocol**	**Representative example**
Blockchain	Public	Proof-of-Work (PoW)	**Ethereum (40):** [PS5, PS12, PS14, PS16, PS19, PS20, PS22, PS23, PS24, PS26, PS27, PS30, PS33, PS39, PS40, PS43, PS49, PS52, PS53, PS57, PS61, PS64, PS65, PS66, PS68, PS70, PS71, PS73, PS75, PS79, PS82, PS83, PS85, PS87, PS88, PS90, PS91, PS94, PS95, PS99]
			Bitcoin (9): [PS2, PS8, PS23, PS35, PS36, PS74, PS80, PS81, PS92]
		Proof-of-Stake (PoS):	**Monax (3):** [PS13, PS40, PS49]
	Private	Byzantine Fault Tolerance (BFT)	**Hyperledger Fabric (15):** [PS6, PS9, PS25, PS28, PS34, PS38, PS44, PS45, PS46, PS50, PS51, PS58, PS93, PS97, PS100]

Continued on next page

Table 2.6 – *Continued from previous page*

Data structure	**Type of blockchain**	**Consensus Protocol**	**Representative example**
		Round Robin	**Multichain (8):** [PS7, PS17, PS37, PS62, PS76, PS77, PS78, PS96]
		Proprietary protocol	**Proof-of- Service (1):** [PS5]
			Proof-of-Inclusion (1): [PS27]
			Proof-of-Authority (1): [PS32]
DAG	N/A	IoTA	[PS11, PS28, PS48, PS60]

Type of blockchain (referring to the use of a public or private blockchain [151]): In a *public blockchain*, anyone can join the network and perform transactions, which could improve transparency, but could lead to user anonymity and data privacy issues. Furthermore, public blockchains have low transaction throughput due to delay in final transaction confirmation, especially in PoW-based blockchains [89]. Most of the systems in the studies use the Ethereum platform to facilitate the deployment of IoT systems supported by the blockchain. These systems include blockchain auditable storage [PS8], hybrid BC-IoT [PS12], privacy SH [PS22], and integrity CPS [PS24]. On the contrary, a *private blockchain* could be managed and hosted by a single organization that defines who can join the blockchain network, thus limiting the number of miners nodes. Furthermore, private blockchains restrict a user with access to only the transactions that correspond to them, allowing competing organizations

to maintain privacy and confidentiality of their transactions, as in the case of Hyperledger [43]. The next set of systems uses a private blockchain (i.e., Hyperledger Fabric and Multichain) to support different IoT use cases as follows: Blockchain for Edge [PS6], IoT updates [PS7], Blockchain for data sharing [PS9], Blockchain as a Service for IoT [PS17]. All of these systems assume that a private blockchain is required to record IoT transactions securely and ensure their privacy. Similarly, a *consortium blockchain* is a hybrid blockchain with public and private blockchain features that is maintained by a group of organizations. Each organization keeps a mining node on the blockchain network and validates a block when the majority of nodes agree on the transaction. Although mining nodes can read all transactions on the blockchain network, this access can be restricted to specific nodes, which could result in the possibility of manipulation due to increased centralization [158].

Data structure (referring to the representation of transactions in the distributed ledger). The data structure consists of a chain of blocks connected to each other, where transactions are stored chronologically [2]. There are two types of data structure for blockchains, called a single chain and multiple chains [150].

- *Single chain:* Use a unique blockchain to record all transactions from different users and business logic [150].
- *Multiple chains:* Use two or more blockchain networks to allow for the separation of concerns between transactions from different users and business logic. For example, a blockchain can be used to store data and another to record access control information [150].

Most studies in the literature use a single chain to record IoT data transactions. For example, [PS22] proposes a blockchain-based architecture to improve data privacy in

smart homes. The architecture consists of IoT gateways that connect a cluster of IoT devices in the smart home to the blockchain, a set of smart contracts to ensure trust access control of IoT data on the blockchain, and a service provider to provide data storage and service recommendation to users. In a similar work [PS13], an out-of-band authentication mechanism for IoT devices using blockchain is presented as a second authentication factor to help authentication of IoT devices. In particular, an out-of-band channel is used as a second authentication method to detect malicious IoT devices and record only legitimate data on the blockchain.

For example, [PS49] allows a distributed architecture to allow opportunistic collaboration between mobile IoT devices, which can share their services and excess computing resources. In particular, IoT devices can create small blockchains based on their location, which can be expanded as more devices are registered on the collaborative network. Smart contracts are deployed in collaborative nodes and the rules are codified to discover services and resources. Similarly, [PS117] propose a framework for secure and efficient IoT data management that consists of a consortium blockchain, side chains, and edge smart devices. The consortium blockchain operates as a control station, while the side chains work as the backbone blockchains for specific IoT scenarios. In particular, off-chain channels are used to connect side chains to the consortium blockchain using a notary mechanism.

Consensus protocols It is the procedure used by all blockchain nodes to reach a common agreement on the state of the distributed ledger. The consensus protocols most commonly used for IoT systems supported by blockchain are described as follows [89, 150].

- Proof-of-Work (PoW): This consensus algorithm requires the blockchain nodes to solve a complex mathematical puzzle to select a miner for the next block generation. This puzzle needs a large amount of computing power and energy, so it is mainly used to

create applications that have strong security requirements (i.e., node identification, authentication, and authorization).

- Proof-of-Stake (PoS): This consensus protocol selects validators (i.e., miners) in a deterministic way based on their stake, where validators with higher coins can be chosen to add new blocks. However, it could lead to a monopoly problem in which an attacker can possess more than 50% of currency and reverse transactions.

- Practical Byzantine Fault Tolerance (PBFT): This consensus algorithm enables the blockchain nodes to share messages among each other to commit a block to the chain. Although it can enhance the security and performance of the blockchain network, it can be expensive due to the number of messages required for consensus.

- Proof-of-Authority (PoA): This consensus protocol selects the authorities (i.e., miners) and assigns them a fixed time slot in the blockchain network to generate blocks. Since PoA operates with a limited number of authorities, the blockchain network can afford to frequent blockchain updates and process more transactions.

The selection of a consensus protocol could have a high impact on the security and scalability of IoT systems supported by blockchain.

Deployment of blockchain

The location of blockchain nodes is a fundamental design decision to consider when deploying blockchain-based systems, as it has an impact on some quality attributes (i.e., performance and scalability) [150, 79]. The most common practice for integrating blockchain and IoT is to deploy the blockchain network on (i) the edge, (ii) the cloud provided by a third party, or (iii) the local network [151, 78]. Most primary studies deploy the blockchain in a local network to minimize latency and guarantee the privacy of IoT transactions. The next set

of studies implements a blockchain in the cloud infrastructure, resulting in high latency and bandwidth consumption. The other set of studies uses an edge platform to deploy a blockchain network where edge nodes can operate as miners and full nodes. However, data computation and storage on edge nodes are still limited and could become a bottleneck in the network as the amount of IoT data increases over time.

2.4 Findings and gaps from primary studies

This section summarizes the most noticeable observations and presents some gaps and opportunities for architecting IoT systems supported by blockchain. In particular, we carefully position our discussion in light of design issues regarding the integration of both technologies from a data perspective.

- *Lack of architectural support for some quality attributes.* Section 2.3 presents the quality attributes most commonly reported in the literature that are necessary to design IoT systems supported by blockchain. In addition to security, performance, and scalability, our analysis reveals that there are other quality attributes, such as interoperability, efficiency, adaptability, and mobility, that can be keys to reason about the dynamism and uncertainties in this category of systems. However, these quality attributes are briefly mentioned in the primary studies and lack architectural support in the literature. For example, IoT devices can join and leave the network at any time, making it easy for attackers to compromise such devices with fake identifiers and manipulate IoT networks in the presence of such dynamic networks [107]. From the blockchain perspective, the business logic encoded in the smart contract variables could require updating according to the IoT context [84]. Therefore, we highlight the need for research and development to provide architectural support for interoperability, efficiency,

adaptability, and mobility in IoT systems supported by blockchain. This gap leads us to formulate RQ2 from Section 2.2.1 and address it in Chapter 3.

- *Lack of focus on the main issues of the blockchain for IoT systems.* Most of the studies in the reviewed literature show that there is a lack of focus on the consensus protocol based on IoT, transaction validation rules, and secure device integration. Therefore, research is required to create consensus protocols oriented to IoT that minimize latency and energy while ensuring the security and privacy of IoT systems. Furthermore, due to the limited memory and storage of IoT devices, they cannot keep a full copy of the blockchain data and mine transactions. Therefore, to improve the scalability of the blockchain and promote the architectural design of the IoT systems supported by the blockchain, IoT systems could take advantage of fog nodes that can preprocess data before sending it to the blockchain. In contrast, there is a lack of IoT-centric transaction validation rules in which IoT transactions can be validated rapidly without causing bottlenecks in the network. A possible solution is the use of off-chain storage to process IoT transactions instead of waiting for block confirmation. This gap led us to formulate RQ2 from Section 2.2.1 and address it in Chapter 3.

- *Lack of focus on the integration of blockchain and IoT from the software architecture perspective.* The systems in the primary studies tend to have little discussion of design decisions to consider when designing IoT systems supported by blockchain. Most of them are designed in an ad hoc manner, and lack of systematic analysis of architectural design alternatives and their impacts on the quality attributes to be satisfied. Only a few studies agree on the allocation and computation of data on-chain or off-chain as the main decisions to be made when architecting IoT systems supported by blockchain [79, 78, 150, 84]. This observation shows that architectural tactics for the integration of blockchain and IoT are still an area for exploration that could greatly benefit software developers and architects. This gap led us to formulate RQ3 from Section 1.3 and

address it in Chapter 4.

- *Lack of focus on the system-level concerns.* The majority of primary studies tend to have a narrow focus on proving that powerful IoT devices can be effectively integrated into blockchain, due to the limited capabilities and connection lifetime of most IoT devices. There are questions related to the integration of both technologies that need to be addressed when systems grow from initial prototypes to operational systems with hundreds of IoT devices, as follows.

  - How do the systems perform when the blockchain network is hosted on the edge, with IoT devices trying to transmit the collected data to the same edge node for pre-processing and blockchain tasks?
  - In the same scenario, what happens when IoT devices lose connectivity to the blockchain network running on edge nodes?
  - How can IoT devices know that the blockchain nodes running on the edge layer are trustworthy, to send transactions to them?
  - In those systems that deploy a blockchain network based on cloud resources, what is the mechanism for ensuring IoT data is protected when it is in transit?
  - What are the tradeoffs between quality attributes promoted by blockchain design configuration and other quality attributes (i.e., network usage, energy efficiency, and latency) that can impact the design of IoT systems supported by blockchain?

- *Lack of large-scale evaluation.* Many systems in the studies use a Proof-of-Concept (PoC) to demonstrate the feasibility of integrating IoT systems with the blockchain, which are implemented on a blockchain test net or local environments [81]. For instance, the consensus protocol in the Ethereum test net (i.e., Kovan and Rinkeby) is Proof-of-Authority (PoA) instead of PoW, which is the de facto consensus protocol in public blockchains. Therefore, the results shown in the evaluation section of the

primary studies are inaccurate and differ in terms of latency from the public Ethereum blockchain. Furthermore, the experiments are run in controlled environments over Wi-Fi connections and with a few IoT devices, which could not reflect real IoT systems with thousands of heterogeneous devices collecting real-time data and transmitting it to the blockchain network. This gap led us to formulate RQ4 from Section 1.3 and address it in Chapter 5.

2.5 Threats to validity

This section summarizes the threats to validity identified in our study and how we deal with them.

External validity: Among the potential external threats in our study, we highlight the fact that we have a limited set of primary studies, which could not represent the state-of-the-art and practices on the architecture of IoT systems supported by blockchain. To mitigate this threat, we applied a search strategy to selected primary studies following the guidelines suggested by Kitchenham and Brereton [69], which was combined with a snowball technique to expand the set of studies collected from the automatic search. We only included peer-reviewed studies (i.e., journals, conferences, and workshops) and excluded non-scientific studies (i.e., blogs, tutorials, etc.) as they do not reliability deliver high-quality scientific contributions. We also defined inclusion and exclusion criteria, which were revised and refined by researchers and experts in the field. Specifically, we discuss the definition of each inclusion and exclusion criterion to have a minimal bias in identifying these primary studies and provide direct evidence of the proposed research questions. It is important to highlight that even when we defined E2 for limiting secondary studies (i.e., surveys and systematic reviews), we considered them to assess the completeness of our set of selected studies and to

identify significant challenges in designing IoT systems supported by blockchain.

Internal validity: We limited the level of influence of extraneous variables in our study by defining a rigorous research protocol, which was developed in consultation with coauthors and other researchers. This protocol describes each stage of the conducted study, including (i) the string search derived from the research questions, (ii) the selection criteria to identify relevant studies, and (iii) the data analysis to extract relevant information from the set of final primary studies.

Construct validity: We performed an automatic search in the largest databases and indexing libraries in computer science and software engineering to collect our primary studies [102, 69]. We also defined a search string using the terms derived from the research questions and their synonyms to identify as many studies as possible to extend the coverage of the automatic search. Additionally, we designed a rigorous and explicit set of inclusion and exclusion criteria to identify primary studies that have direct evidence of research questions. We ensured the validity of the primary studies collected by performing an automatic search in multiple well-known scientific databases and indexing libraries in computer science and software engineering [102, 69]. We did not restrict our search of primary studies to the publication date to extend the coverage of the automatic search. As some studies lack architectural definition, we performed a title, abstract, and full-text reading to reduce misinterpretation in the selection process.

Conclusion validity: We mitigated the possible threats to the relationship between the extracted data and the results obtained by applying a well-defined and rigorous search protocol, which was defined following the most recent guidelines on systematic mapping studies [102, 69]. We also reviewed and refined the protocol with experts in the field to ensure its completeness and applicability. This work applied qualitative and quantitative analysis to describe the results of our study in terms of the proposed research questions

and used the extracted data for further analysis (i.e., quality tradeoffs, constraints, and dependencies among tactics, etc.). We document each stage of our study to facilitate its understanding and replication by independent researchers.

2.6 Related Work

In this section, we present relevant studies on the adoption of blockchain in IoT systems and fundamental work on IoT systems supported by blockchain. Our review devotes the most attention to work closely related to the identification of the quality attribute requirements and design decisions necessary for the architecture of this category of systems. There have been several studies that have attempted to analyze blockchain as a potential technology to solve security issues in IoT systems. Unfortunately, most of these studies assess the integration of blockchain in IoT systems from an application perspective without considering their architectural design from the data perspective. Our work differs mainly from existing studies on the integration of blockchain in IoT systems as follows. First, we conduct an SLR to investigate the commonly reported quality attributes and design decisions to be considered when designing IoT systems supported by blockchain. Our findings are drawn from 100 research papers selected from a set of 575 relevant publications on IoT and blockchain. Second, we focus particularly on the categorization of architectural design decisions that have been made to satisfy the quality attribute requirements. Third, we identify potential areas for future research that include architectural support for specific quality attributes, empirical research to evaluate the impact of the identified quality attributes, and research effort to explore tradeoffs among the quality attributes and design decisions.

2.6.1 Surveys in IoT and blockchain

Recently, Conoscenti et al. [25] conducted a comprehensive systematic review of the literature to study the application of blockchain technology and its benefits in terms of decentralization and security. The study describes several use cases where data storage management, trade of goods and data, and identity management have been identified as potential IoT cases to be enhanced with blockchain. Furthermore, Christidis et al. [23] emphasized the advantages and disadvantages of adopting blockchain in IoT systems and the use of smart contracts for data sharing and autonomous governance. Yeow et al. [155] critically reviewed the decentralized consensus systems to architect edge-centric IoT systems focusing on the data structure, consensus protocols and transaction models. In addition, Fernández-Caramés et al. [37] presented a review of the impact of blockchain in IoT and current challenges with respect to the design, development, and deployment of IoT systems supported by blockchain. This review also identifies gaps in the literature that can guide researchers and practitioners in the design of future IoT systems supported by blockchain. Reyna et al. [111] discussed the benefits and challenges of the integration of blockchain and IoT and recent platforms and applications for combining these technologies. This survey also presents three architectures to facilitate the communication between IoT devices and the blockchain. Furthermore, Ali et al. [2] presented a comprehensive survey to investigate current efforts for the integration of blockchain and IoT and summarize some solutions to improve data privacy, security, identity management, data management, and monetization in IoT systems. Similarly, Panarello et al. [98] conducted a systematic survey to show current research efforts on the use of blockchain in IoT applications by categorizing the existing literature based on different domains. Furthermore, the survey describes the challenges and future research directions to realize the adoption of blockchain into IoT systems. In another work, Ferrag et al. [38] presented a survey on current efforts, trends, and challenges in the integration of blockchain in IoT systems by providing an overview of the use of blockchain in different IoT domains (i.e.,

Internet of Vehicles, Internet of Energy, Edge Computing). Hong-Ning et al. [27] conducted a survey on IoT and blockchain with a special focus on the challenges in IoT, an overview of blockchain technology, and the main opportunities to integrate both technologies. In particular, the authors summarize the main IoT applications supported by blockchain and the key role of 5G beyond networks in the convergence of IoT and blockchain. Furthermore, Mingli et al. [145] proposed a systematic survey of blockchain and its application in the IoT, where fundamental issues and open challenges are discussed about the integration of both technologies. In particular, the authors study the blockchain architecture with a special focus on the adoption of blockchain in other areas (i.e., Artificial Intelligence and Edge Computing). In contrast to the above, our study explicitly defines what the commonly reported quality attributes in the literature are considered in the design of IoT systems supported by blockchain. Sin Kuang et al. [81] presented solutions for the integration of blockchain and IoT. Although most surveys primarily focus on the advantages of integrating blockchain and IoT in terms of decentralization, security, and data privacy, our findings are based on 100 research publications on blockchain and IoT to identify the common quality attributes and design decisions reported in the literature that must be satisfied when integrating these two technologies.

2.6.2 Fundamental work on the integration of blockchain and IoT

Lee et al. [72] presented a secure and scalable firmware update scheme based on blockchain, where IoT devices first need to calculate the hash of the downloaded file to check its integrity. To reduce the computational load and data storage requirements on the blockchain, the system relies on a P2P network where firmware updates are distributed over multiple nodes to ensure their availability. Moreover, Dorri et al. [31] proposed a lightweight blockchain with a three-layer architecture: smart home (centrally managed), overlay network (public

blockchain) and cloud to improve the security and privacy of smart homes. The system implements a distributed trust model in the overlay network to reduce the processing overhead and energy requirements of the PoW consensus. An alternative way is to rely on edge computing to shift computation and data storage requirements to powerful IoT devices to minimize latency and improve the scalability of the blockchain network, as suggested by Stanciu [128]. Similarly, Bahga et al. [7] presented a trusted and decentralized platform called BPIIoT that enables powerful devices to communicate and manage manufacturing resources in a P2P network. This system relies on an intermediary component acting as a one-to-one proxy to facilitate communication between the IoT nodes and ensure communication between them. A different approach is suggested by Shabandri et al. [120] where the system relies on Tangle structure instead of blockchain to improve scalability and reduce latency in transaction confirmation. Only a few attempts have been identified in the literature on the integration of blockchain and IoT from the perspective of software architecture. Among the existing works, Liao et al. [79] proposed a taxonomy to capture the most significant architectural issues in blockchain-based systems and their impact on non-functional requirements. Similarly, Liao et al. [78] identified the architectural design issues for architecting IoT systems supported by blockchain that include location of the blockchain nodes, the distribution of logic and data, and the integration mechanisms. Based on these design decisions, this study proposes four architectural styles: fully centralized, pseudo-distributed, distributed, and fully distributed. In another work, Reyna et al. [111] emphasized three alternatives to enable interaction between IoT devices and the blockchain, including the IoT-IoT, IoT-blockchain and hybrid approach. Xu et al. [149] described a set of architectural patterns for blockchain-based applications that include *External world patterns, data management patterns, security patterns, and contract structural patterns.* Similarly, Eberhardt et al. [34] proposed five patterns regarding on-chain or off-chain data called *Challenge response pattern, Off-chain signatures pattern, Content-addressable storage pattern, Delegated computation pattern, and Low contract footprint pattern.*

2.7 Conclusion

This chapter presented the results of an SLR to investigate the common quality attributes and architectural design decisions in IoT systems supported by blockchain in the context of RQ1. The main objective of this question was to identify the design decisions made in the literature that can satisfy the quality attribute requirements of this category of systems. To this end, we examine 100 primary studies and classify them in terms of the quality attributes they support and the design decisions they made. The results revealed that security, scalability, performance, and interoperability are the common quality attribute requirements reported in the literature and must be considered in the design of current and future IoT systems supported by blockchain. Furthermore, we identified design decisions related to data storage and computation, blockchain scope, consensus protocol, blockchain data structure, and blockchain deployment and investigated how they influence the achievement of quality attribute requirements.

In addition, this analysis also allowed us to identify gaps and opportunities for our work in the remainder of this thesis, as follows: (i) besides security, scalability, performance, and interoperability, there are other quality attributes that are relevant to the operation of IoT systems supported by blockchain, such as adaptability and mobility, which lack architectural support in the literature; (ii) investigation is required to define the architectural tactics to support this category of systems, as well as evaluate how they can meet the desired quality attributes; (iii) additional research is needed to explore the architectural styles proposed in the literature that can guide the development of the systems. This work attempts to address the last three, which will facilitate the development of IoT systems supported by blockchain to become widely adopted in academia and industry.

The next chapter presents a catalog of architectural tactics extracted from architectural design decisions made in primary studies. This set of tactics can provide architects

and developers of IoT systems supported by blockchain with different options to satisfy the quality attribute requirements of the system.

Chapter Three

Architecting IoT systems supported by blockchain: A Catalog of Tactics

In chapter 2, we present the common software quality attributes, architectural tradeoffs, and design decisions necessary for the development of IoT systems supported by blockchain that were reported in the SLR. In this chapter, we codify a catalog of architectural tactics derived from the architectural design decisions identified in the primary studies (thereby addressing RQ2). Tactics are design decisions that aim to satisfy a particular quality of the system. We use the architectural pattern language to describe the architectural tactics and ease their adoption when architecting IoT systems supported by blockchain. Finally, we use the design science-based evaluation approach to reflect on the catalog of architectural tactics.

3.1 Overview

Architectural tactics are design decisions to achieve a particular quality attribute of interest and can be composed into a software architecture design [105]. Using tactics, architects can select among alternatives (i.e., tactics) to deliver candidate architectures for complex software

systems, such as IoT systems supported by blockchain. These systems are characterized by the IoT constraints (i.e., high mobility, high velocity, and high data volume) and blockchain inherent limitations (e.g., limited computational power and data storage) [147]. Given the complexity of this category of systems, it would be of great value for software architects and designers to have a set of tactics that can be used to achieve particular qualities of the system.

Despite the hype about the application of blockchain in IoT systems [146, 157, 30, 123, 56, 28, 74], only a few researchers have attempted to investigate common design decisions that can drive the development of IoT systems supported by blockchain to achieve desired quality attributes [79, 151, 150, 111]. However, these approaches systematically examine design issues related to the integration of blockchain and IoT, without mentioning architectural tactics. Therefore, there is a general absence of a comprehensive investigation and body of architectural knowledge documenting the architectural tactics that can be used to build candidate architectures for IoT systems supported by blockchain that achieve particular quality attribute requirements.

This chapter presents the architectural tactics for IoT systems supported by blockchain that have been extracted from the common design decisions across the primary studies. First, we narrow down the commonly discussed architectural design decisions among the primary studies identified. In particular, we selected the studies by looking at (i) explicitly stated quality attributes, (ii) inferred quality attributes from the literature, and (iii) commonly identified components and their relations. Next, we codify the identified design decisions in a catalog of tactics that can be reused in the development of this category of systems. This catalog can guide software architects and designers in the architectural design of software architectures for IoT systems supported by blockchain that meet the intended qualities and cope with the IoT constraints and the inherent limitations of the blockchain. We document the architectural tactics using the architectural pattern language used in [73] to facilitate their adoption

in IoT systems supported by blockchain. Finally, we use the design science-based evaluation approach to reflect on the catalog of architectural tactics.

It is worth noting that the identified tactics are not exhaustive. Instead, we attempt to provide a categorization of existing tactics in the literature to guide architects and researchers in the inception and implementation of specialized architectures for IoT systems supported by blockchain and to improve their decision-making options. Such catalogs (e.g., [12, 20]), have already shown their value in helping practicing architects in both design and analysis. We perform a qualitative evaluation of the tactics by listing the primary studies in which it is applied as an example. We also observe gaps for future research (i) investigation is required to evaluate the impact of the architectural tactics in this category of systems, and (ii) additional research is needed to explore the trade-offs among the quality attributes and identified tactics. These opportunities for future research require extensive collaboration between industry and academia to implement, deploy, and evaluate architectural tactics and control quality attributes in large-scale IoT systems supported by blockchain. This chapter presents the following.

- A catalog of relevant architectural tactics for IoT systems supported by blockchain derived from the SLR results to assist software architects and developers in the choice of reference architecture styles to build a software architecture that meets system qualities.
- A categorization of the identified tactics for security, scalability, performance, and interoperability as key quality attributes that drive the development of IoT systems supported by blockchain.
- A set of potential areas for future work that include empirical research to evaluate the impact of identified architectural tactics on IoT systems supported by blockchain and a research effort to explore trade-offs among the quality attributes and identified

tactics.

The remainder of this chapter is organized as follows. The section 3.2 presents the architectural tactics for the design of IoT systems supported by blockchain. Section 3.4 concludes the chapter.

3.2 Architectural tactics for IoT systems supported by blockchain

This section presents the extracted architectural tactics for the design of IoT systems supported by blockchain. First, we highlight the difference between architectural tactics and patterns to provide software architects with an in-depth understanding of their impact on the architecture design of a system. According to Harrison and Avgeriou [50], *architectural tactics* are "design decisions that influence the control of individual quality attribute requirements", while *patterns* "describe the high-level structure and behavior of a software system as the solution to recurring problems". For example, a design decision concerning security could be how to prevent attacks on the system. A possible tactic to improve security could be the authentication of users [50]. It is worth noting that this work focuses on the extraction of architectural tactics from the reviewed literature to satisfy particular quality attributes of IoT systems supported by blockchain and to provide different options for the architectural design of this category of systems. In this context, Bass et al. [12] present a list of architectural tactics to meet the following quality attributes: availability, interoperability, modifiability, performance, security, testability, and usability. We examine whether these tactics have been applied or adjusted in the context of blockchain and IoT systems to provide a catalog of relevant architectural tactics for IoT systems supported by blockchain. Our work elicits the tactics from the primary studies based on (i) the explicitly stated quality attributes, (ii)

inferred quality attributes from the primary studies, (iii) the commonly reported blockchain-based design decisions, and (iv) common components and their relations across the selected studies. Specifically, we pair a quality attribute with relevant design decisions and translate them into architectural tactics. In addition, we rely on a surrogate component widely used in cyber-foraging systems to offload computation or data to more powerful devices [73]. In most proposed tactics, the surrogate acts as an intermediate between IoT devices and the blockchain and is used to collect sensor readings from resource-constraint devices, which cannot directly to a blockchain network and perform mining tasks. In other cases, IoT networks comprise powerful devices that can connect directly to the blockchain without the need for a surrogate component. Table 3.1 shows a catalog of architectural tactics for security, scalability, performance, and interoperability. Figure 3.1 shows the identified tactics. We report each tactic using the template described by Lewis and Lago [73] as follows:

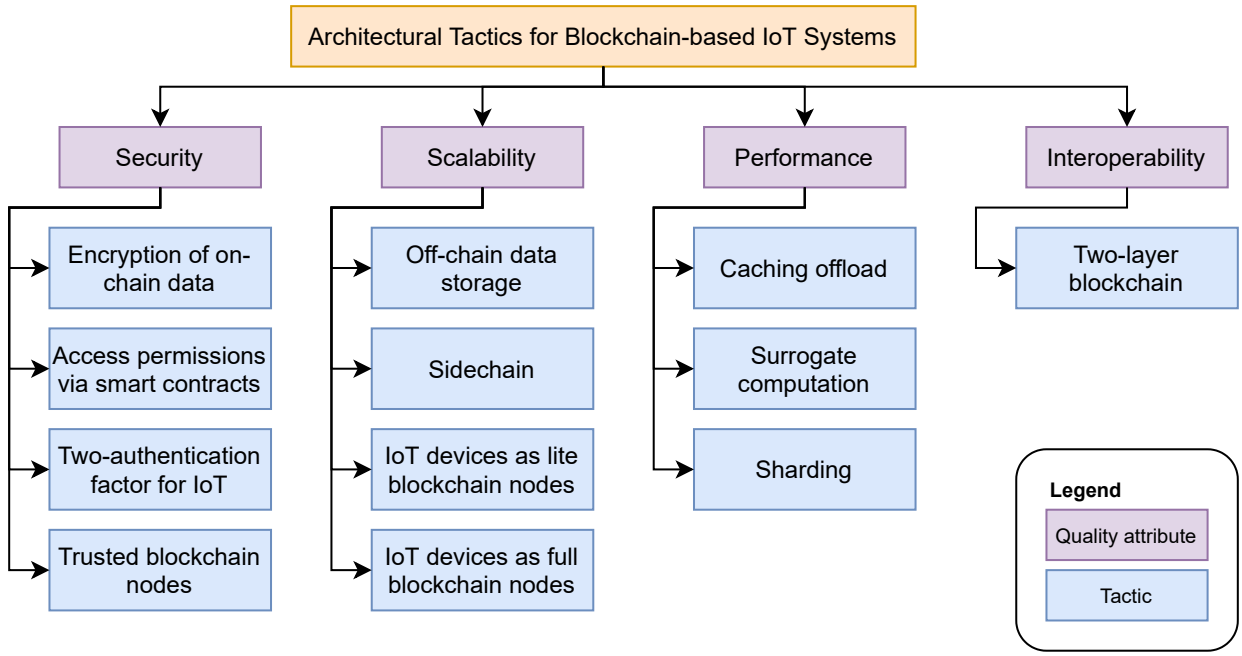

Figure 3.1: Architectural tactics for blockchain-based IoT systems.

- *Summary:* Brief introduction of the tactic.
- *Motivation:* Rationale behind the implementation of the architectural tactic.

- *Description:* Detailed explanation of the components of a tactic and their interaction to achieve a particular quality attribute.
- *Constraints:* Benefits and drawbacks of applying the tactic.
- *Example:* Application of the tactic in the existing literature.
- *Related tactic:* Relation with other tactics to achieve its potential.
- *Variations (optional):* Slight modification of the tactic from its original form to optimize it.

Although the same diagram style was used to describe most of the architectural tactics, slight modifications were required to understand some of them.

Table 3.1: Architectural tactics for IoT systems supported by blockchain.

Quality attribute	**Tactic name**	**Description** [0.5ex]
Security	(1) Encryption of on-chain data	Encrypt IoT data before sending transactions to the blockchain to ensure its confidentiality and privacy.
	(2) Access permission via smart contracts	Enable access control to IoT data through smart contracts.
	(3) Two authentication factors for IoT devices	Enable an additional layer of security to authenticate IoT devices.

Continued on next page

Table 3.1 – *Continued from previous page*

Quality attribute	**Tactic name**	**Description** [0.5ex]
	(4) Trusted blockchain nodes	Ensure integrity of data and IoT devices by identifying and authenticating them in the blockchain network.
Scalability	(5) Off-chain data storage	Use a third-party offline data storage for IoT raw data, while keeping a digital hash of critical data on-chain for verification.
	(6) Sidechain	Improve scalability of the blockchain by relying on child chains connected to a parent chain.
	(7) IoT devices as lite blockchain nodes	Connect resource-constraint IoT devices to the blockchain network through powerful IoT devices.
	(8) IoT devices as full blockchain nodes	Use powerful IoT devices as full blockchain nodes.
Performance	(9) Caching offload	Use a cache system to offload a subset of data and make further data requests faster.
	(10) Surrogate computation	Delegate computation-intensive tasks to edge servers to reduce computation and data storage load in blockchain nodes.

Continued on next page

Table 3.1 – *Continued from previous page*

Quality attribute	**Tactic name**	**Description** [0.5ex]
	(11) Sharding	Increase transaction throughput confirmation in blockchain networks.
Interoperability	(12) Two-layer blockchain	Enhance interoperability of public and private blockchains by introducing a two-layer blockchain architecture.

3.2.1 Encryption of on-chain data

Summary: Encrypt IoT data before sending it as transactions on a blockchain to ensure their integrity and immutability.

Motivation: One of the main issues in public blockchains is the lack of privacy, since anyone on the Internet can join the network without permission [151]. As a result, all transactions on the blockchain are available to everyone on the network, and almost every participant has a copy of the entire chain [150]. Therefore, IoT data cannot be deleted or altered on the blockchain network, which leads to better transparency and auditability, but impacts privacy and confidentiality in IoT systems.

Description: Figure 3.2 shows the main components of the encryption of the on-chain data tactic. This tactic requires encrypting IoT data to enhance its security before replicating it across the blockchain nodes. A possible way to encrypt and decrypt data using asymmetric cryptography is described below [89]. First, one of the nodes in the blockchain creates a public key and shares it during an initial key exchange. Next, if a user wants to send data to a blockchain, he encrypts the data with the public key of the participant who is allowed to view the data. The participant in possession of the corresponding private key can then decrypt the data.

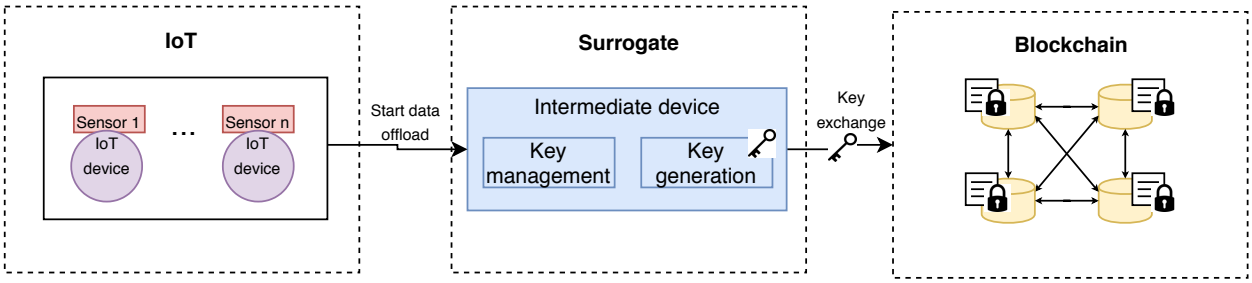

Figure 3.2: Encryption of on-chain data, where a surrogate device handles the encryption key.

Consequences:

Benefits:

- *Confidentiality:* IoT data on a public blockchain is not in plain text, instead it is encrypted with the public key of the authorized participant in a blockchain network and accessible only using the corresponding decryption key.

Drawbacks:

- *Key management and sharing:* The encryption and decryption keys must be securely

shared off-chain and distributed among authorized nodes before submitting any IoT data to the blockchain. If key management is not handled carefully or shared on a public blockchain, encryption keys could be compromised and disclosed. This results in a lack of confidentiality and integrity of IoT data stored in a blockchain.

- *Access permission:* Once IoT data has been stored in a blockchain, it is difficult to revoke read access, as the blockchain ensures immutability. Thus, a participant in a blockchain network can access encrypted data as long as he is in possession of the corresponding decryption key.

- *Data immutability:* Even when IoT data recorded on a blockchain remain encrypted, it could be subject to brute-force decryption attacks [149]. With the advancements in quantum technology, current encryption algorithms could become ineffective in the future [68].

Related tactic: Off-chain data storage tactic (Section 3.2.5).

Example:

- *Optimized blockchain* [PS2]. All transactions performed on the IoT network are signed and encrypted before being sent to a blockchain and becoming available to all blockchain nodes.

- *IoT updates* [PS7]. The system relies on asymmetric encryption using RSA keys for updating signing and encryption to guarantee data confidentiality and integrity of IoT transactions.

- *Blockchain auditable storage* [PS8]. The transactions consist of the ownership of data streams and corresponding access permissions, and are encrypted using asymmetric

cryptography to guarantee data confidentiality and integrity.

- *BLE-IoT* [PS71]. The gateway encrypts user preference for IoT devices and stores it in the blockchain to ensure its privacy and confidentiality.

- *IoST* [PS77]. Data requests sent to the rule-based expert system are encrypted using the synchronous AES encryption method before being sent to a blockchain for immutable storage.

- *IoT privacy* [PS80]. IoT devices manage a public and private key to send encrypted sensor readings to a validator node, which logs the received data as data creation events before adding them as encrypted transactions to the side chain.

- *IoT data assurance* [PS18]. The data collected by drones is encrypted and signed using a public and private key pair to protect its integrity before making it available in the blockchain network. It is accessible only to whoever owns the corresponding decryption key.

- *P2P data monetization* [PS28]. The system uses credentials (i.e., certificates and keys) to protect all messages on the IoT network before recording them on the blockchain.

- *Blockchain Lightweight IoT Clients* [PS30]. Transactions consist of modifications to the account states and are signed using asymmetric cryptography and identified by their hash value, as described in the Bitcoin specification.

- *Emergency SH* [PS33]. Asymmetric encryption is used to protect user data and sensitive information from malicious users on the network during data transit or at rest. RSA asymmetric encryption of key length 1024 bits is used to sign the data before pushing them to the blockchain.

3.2.2 Access permission via smart contracts

Summary: Enable access control rights to IoT transactions and execute automatic tasks based on predefined conditions using smart contracts.

Motivation: Due to the limited computational, storage, and power in sensors and embedded devices, IoT systems rely on cloud services for data processing and analysis. However, data in the cloud can be manipulated and altered by cloud providers [111]. Therefore, blockchain with its smart contracts empowers users with control over their data by restricting access only to authorized blockchain nodes without relying on cloud service providers [71].

Description: Figure 3.3 shows the main components of this tactic. The access permission via the smart contract tactic requires the deployment of smart contracts on the blockchain network to grant access to IoT data or perform arbitrary computations. Smart contracts can encode fine-grained permissions or contextual policies for sharing services and resources and run as part of transactions autonomously [2]. When a user wants to access a protected resource, he has to encrypt the transaction with his public-private key pair and send it to the address of the smart contract on the blockchain. Next, the execution of certain operations in a transaction can be restricted to certain authorized blockchain nodes to enhance the security of IoT systems.

Consequences:

Benefits:

- *Security:* Only the blockchain nodes authorized by the smart contracts can access the

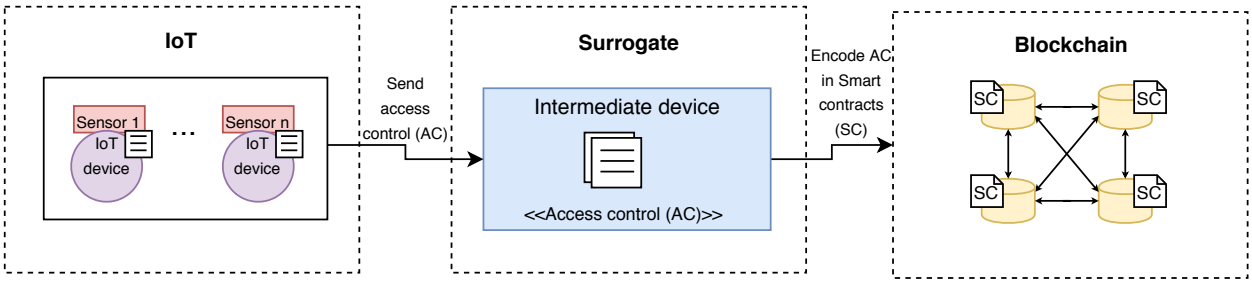

Figure 3.3: Access control via smart contract, where the surrogate handles IoT permissions.

user records, without the need for a trusted third party or a cloud service provider for validation or authorization.

Drawbacks:

- *Cost:* If a public blockchain is used to store a smart contract, the implementation of access control permissions is extra cost. This cost includes the implementation and execution of the smart contract on the blockchain network, as each blockchain node must validate it before approval [89].
- *Flexibility issues:* If access control is not considered initially, it could be difficult to introduce it afterward due to the structural immutability of smart contracts. Thus, the implementation of access control via smart contracts can help to deal with changing requirements in the system as long as it is acceptable to have those changes documented as immutable transactions.
- *Codification issues:* The smart contracts should be well-written since once deployed, data stored on them cannot be modified which can lead to loss of money, wrong decisions, and catastrophic consequences in IoT systems [78].
- *Deployment issues:* Ethereum and Hyperledger Fabric are the most popular blockchain platforms that support smart contract implementation [113]. However, there are other

blockchain platforms that also facilitate the implementation of smart contracts such as EOS, Cardano, Stellar, and NEO [89, 140].

Related tactic: N/A

Example:

- *Blockchain manufacturing* [PS76]. The system uses smart contracts to create agreements between users and service providers. These rules are encrypted using symmetric encryption and ensure that only authorized users can use services on the network.
- *Bubble of trust* [PS79]. The system uses a smart contract to create agreements between users and service providers, which are encoded as encrypted rules. These rules could be used to ensure that only authorized users can use services on the network.
- *Blockchain as a service for IoT* [PS17]. The system implements smart contracts to grant access only to authorized participants in the blockchain network who own the required key. They can download and decrypt the protected resources from the blockchain network.
- *MeDShare* [PS4]. Smart contracts are deployed to enable access control policies, data sharing, and revoke access to health data to enhance its security and privacy. Additionally, health data provenance and auditing are ensured since cloud service providers maintain a blockchain network to ensure immutability and data integrity.
- *Privacy SH* [PS22]. A scalable and decentralized access management mechanism is implemented for IoT systems using blockchain. Due to the limited capabilities of most IoT devices, they are connected to a manager node, acting as a lightweight node in the blockchain network. This node defines access control permissions as transactions

that are encoded in a single smart contract and executed by the P2P network to make them accessible to all blockchain nodes.

- *Blockchain meets IoT* [PS75]. The system deploys three smart contracts (i.e., access control contract, judge contract, and register contract) to manage access control to IoT records stored in the blockchain network. Smart contracts allow one to register, remove, and update misbehavior-judging methods to manage access control policies. They empower users with control over their data and facilitate data sharing among trusted participants in the blockchain network.

3.2.3 Two-authentication factor

Summary: Enable an additional security layer in the IoT device authentication process to ensure the integrity and confidentiality of sensor data.

Motivation: With the growing number of IoT devices and the large amount of sensitive and critical data collected by them, security and data privacy become key concerns in IoT systems [5]. However, it is not possible to implement complex security protocols (i.e., encryption and authentication) in IoT devices due to their limited computation, storage, memory, and power lifetime [95]. Although blockchain is expected to solve security issues in IoT systems, there is still a need to develop authentication schemes to protect IoT data in transit where an adversary can take advantage of IoT devices and launch physical or side-channel attacks [10].

Solution: Figure 3.4 shows the main component of this tactic. Implementing a two-factor authentication mechanism in a resource-constrained IoT device requires the use of an out-of-band channel instead of passwords and shared secret keys [144]. The out-band channel

is provided by manufacturers as the first authentication method. This tactic operates as follows. The relationship in terms of proximity and device message exchanges between IoT devices and their verifier is stored in the blockchain network. If a device is moved beyond its designated verified location, it is automatically detected and treated as a malicious outsider, as the distance relationship is already stored as an immutable transaction in the blockchain. It ensures that only authorized IoT devices can be connected to the verifier and that only their transactions can be recorded on the blockchain network.

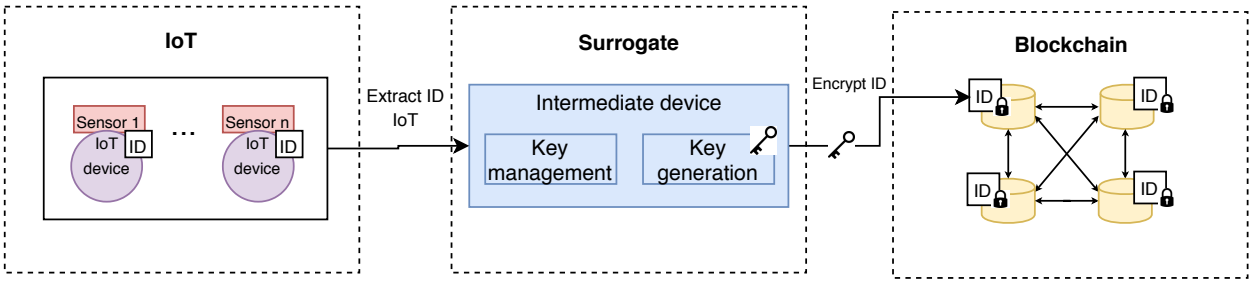

Figure 3.4: Two authentication factors, where the surrogate records proximity and IoT message exchange.

Consequences:

Benefits:

- *Integrity:* Implementing a two-factor authentication mechanism improves the security of IoT devices and protects sensitive and critical user data from malicious actors. This also ensures that only authorized devices can send transactions to the blockchain, guaranteeing the integrity of IoT devices and on-chain data and preventing blockchain nodes from overloading.

Drawbacks:

- *Communication complexity:* The use of low power wireless communication and heterogeneous hardware from IoT devices makes it difficult to create the relationship between an IoT device and its verifier and ensure the integrity of proximity information between them.

Related tactic: N/A

Example: An example of the application of the two-authentication factor has been identified in IoT authentication [PS13]. This system uses wireless channel characteristics to distinguish between home IoT devices and outside devices. Each device needs to authenticate against the verifier device to access services and data at home. The relationship between the device and its verifier is recorded on the blockchain, which makes it easier to detect when an adversary IoT device wants to gain access to the house.

3.2.4 Trusted blockchain nodes

Summary: Ensure the integrity of sensor data and IoT devices by identifying and authenticating them in the blockchain network.

Motivation: The heterogeneity and dynamic connection of IoT devices (e.g., devices can join and leave the network) make it difficult to assign an ID to identify devices in the IoT network [89]. Before IoT data is sent to the blockchain, its integrity is mainly dependent on the security of IoT devices. However, due to their limited computation, storage, and connection lifetime, most IoT devices are vulnerable to attacks that can compromise devices with fake identifiers to join the network [107].

Description: Figure 3.5 shows the main components of this tactic. The trusted blockchain nodes tactic relies on the creation of zones based on the geographical location of IoT devices where they can trust and authenticate each other [48]. A zone consists of a group of IoT devices managed by a master entity, where each device called a follower is provided with a ticket that includes (1) group ID (i.e., a bubble that belongs to), (2) object ID (i.e., follower's identifier), and (3) a pubAddr (i.e., follower's public address). Any device outside the zone is considered malicious and therefore cannot send transactions to the blockchain network. In particular, the smart contract verifies the identity of the followers and the validity of the ticket to associate it with a bubble.

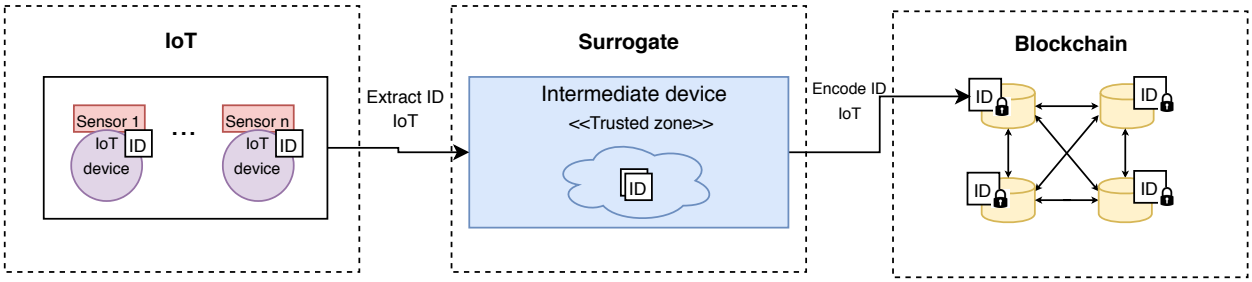

Figure 3.5: Trusted blockchain where the surrogate supports the trusted IoT zones.

Consequences:

Benefits:

- *Integrity:* The creation of trusted zones ensures the integrity of the device, while all devices outside of a zone are considered malicious and data from them are not sent to the blockchain.
- *Identity management:* The identification of IoT devices facilitates the implementation of access control policies and authentication mechanisms to ensure the security of IoT systems supported by the blockchain.

Drawbacks:

- *Compromised trusted authority:* A trusted authority is required to authenticate and identify IoT devices as blockchain nodes, but it could become a bottleneck and a single point of failure in the network.

Related tactic: N/A

Example: This tactic has been identified in Bubbles of Trust [PS79] that creates virtual zones to enable secure communication between devices and consider devices that do not belong as malicious. This approach requires a master entity acting as a certification authority to allow followers (i.e., IoT devices) to participate in the virtual zone and sends transactions to the main blockchain to create a zone at the blockchain level.

3.2.5 Off-chain data storage

Summary: Use offline data storage to record raw IoT data, while keeping a digital hash of the data on-chain for verification.

Motivation: Due to the growth in the number of IoT devices, a large amount of data is generated in real or near real-time that needs to be analyzed and stored securely to protect it against cyber attacks [5]. However, the blockchain has limited computation and data storage, restricting the number of transactions to be recorded on-chain [150]. In addition, the use of public blockchains costs money and could even be more expensive than traditional storage solutions. For example, Ethereum manages a block gas limit to determine the number, computational complexity, and data size of the transactions included in a block [89].

Solution: Figure 3.6 shows the main components of the off-chain data storage tactic, which consists of IoT devices, a surrogate, an off-chain data storage, and a blockchain. The basic functioning of this tactic is described in the following [149, 84]. First, the data collected by the IoT devices is sent to the surrogate, which acts as a gateway between the IoT devices and the blockchain. Next, the surrogate processes the IoT data and decides which data should be recorded on the blockchain or in an off-chain data storage (i.e., private cloud, a local database, or peer-to-peer storage, such as IPFS [1] or Storj [2]). A common practice is to store raw IoT data in off-chain data storage and keep critical data or hashes of the data on the blockchain (i.e., on-chain) [150]. Thus, this tactic improves the performance of the blockchain by relieving on off-chain data storage of raw data. It is important to mention that a surrogate device cannot act as off-chain storage due to its limited computational and power resources, compared to off-chain data storage with large computational and storage capabilities to perform blockchain tasks (i.e., transaction processing, mining).

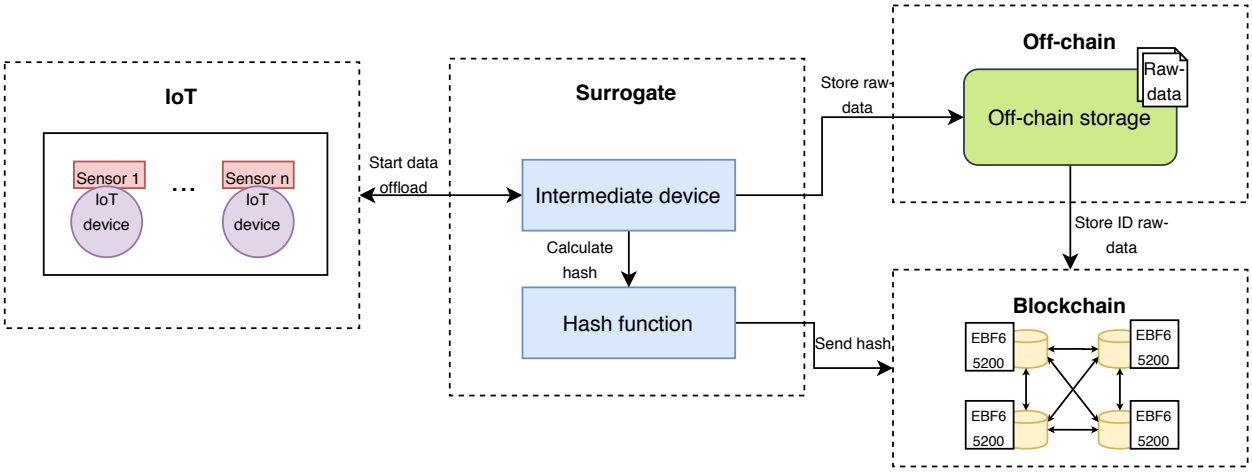

Figure 3.6: Off-chain data storage, where a surrogate manages IoT raw data and calculates its hash.

Consequences:

[1] https://ipfs.io/

[2] https://www.storj.io/

Benefits:

- *Integrity:* To check the integrity of the off-chain data, it is possible to compare the hash of the IoT data stored on-chain with the one generated from the raw data recorded off-chain.
- *Cost:* Since the use of on-chain storage has a high cost, all transactions sent to the blockchain can be summarized in a hash to reduce this cost.
- *Data immutability:* Since the identifier of raw data is stored on-chain, any change to the off-chain data can be detected if the interested parties have access to the off-chain data.

Constraints:

- *Privacy:* The blockchain cannot ensure data privacy, increasing user concerns about data manipulation and loss of information.
- *Data loss:* Since raw data is stored off-chain, it could be deleted, altered or manipulated by service providers in the cloud, and only its identifier could remain immutable on-chain.
- *Data sharing:* While on-chain data can be securely shared among authorized blockchain nodes through smart contracts, off-chain data requires new approaches to data management.

Related tactic: Encryption of data (Section 3.2.5).

Examples:

- *Blockchain auditable storage* [PS8]. The raw data collected by IoT devices is stored off-chain and only its identifier (i.e., hash pointer) is recorded on-chain to ensure its integrity and confidentiality.

- *Blockchain for data sharing* [PS9]. The system summarizes a set of all transactions to be recorded in the blockchain in a digital fingerprint (i.e., hash), which ensures data integrity and transparency. If the integrity of the off-chain data needs to be verified, a hash of the raw data located off-chain can be generated and compared with the hash of the on-chain data.

- *IoT protection-blockchain* [PS14]. A hash of the raw data is generated and stored on-chain to reduce the cost of on-chain data storage on public blockchains. In addition, this hash of the on-chain transaction is recorded in a local database to verify the integrity of the raw data.

- *IoT data assurance* [PS18]. A set of IoT records are compressed using a hash function to get a unique identifier (i.e., hash), which is stored on-chain and off-chain (i.e., cloud solution) to enhance the transparency and auditability of IoT data.

- *IoT exchange* [PS20]. The systems distinguish between two types of data: device data and exchange data. The former can be stored in a local database, a cloud database, or even a wireless sensor network managed by the owner, while the latter is used to track the data exchange process. In particular, a hash is generated from the IoT raw data and recorded on-chain to verify its integrity.

- *IoT privacy* [PS80]. A decentralized access control model with built-in privacy is proposed, where an Interplanetary File System (IPFS) server is used to group and replicate IoT data on the P2P network without the need for a third party. The hashes of the IPFS files are recorded on-chain through smart contracts, and the access control permissions for on-chain data are stored off-chain.

3.2.6 Side chain

Summary: Improve the scalability of blockchain by relying on a chain that is attached to the main chain using a two-way peg.

Motivation: Due to the increasing amount of data generated by IoT devices, extensive computation and large storage space are required to process and record IoT data securely. However, blockchain still has limited computing and data storage resources, which places some restrictions on the adoption of blockchain in IoT systems.

Description: Figure 3.7 shows the main components of this tactic, including IoT devices, side chain, and blockchain. A side chain is an additional blockchain known as a child chain connected to the original blockchain known as the main chain or the parent chain via a two-way peg. The two-way mechanism enables the exchange of tokens and assets between the main chain and the child chains, and vice versa, at a predetermined rate. The basic functioning of this tactic is as follows, assuming that IoT devices have enough computational power and battery lifetime to act as full blockchain nodes [228]. Firstly, an IoT device in the parent chain can lock their assets by sending them to an output address, and thus the IoT device cannot spend them. After the transaction is completed, a confirmation is sent across the side chains along with a waiting period for additional security. Finally, once the waiting period is over, the same number of locked assets are delivered on the side chain, allowing the IoT device to access and use them. The same process is executed when assets are exchanged from a side chain to the parent chain, where each side chain has its miners to validate transactions and periodically report back to the parent chain to update its status. Thus, side chains can remove bottlenecks on the parent chain and increase the speed and scalability of the entire network, as well as allowing separation of concerns among various use cases.

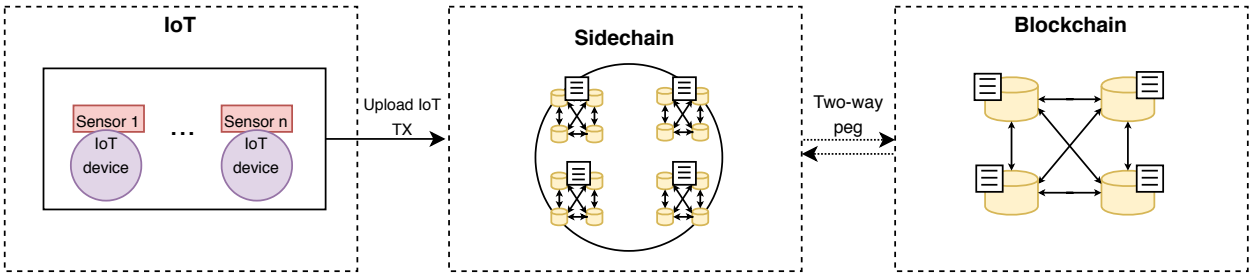

Figure 3.7: Side chain connected to the main blockchain.

Consequences:

Benefits:

- *Scalability:* The use of side chains improves the scalability of the main blockchain, as IoT transactions are computed using surrogate chains.
- *Interoperability:* Each IoT application can run in a side chain and securely exchange digital assets with other surrogate chains at a predetermined rate based on the IoT application requirement.
- *Security:* Each side chain defines its level of security and consensus protocol. If a participant in a surrogate chain acts maliciously, the transactions in the other surrogate chain or on the main blockchain cannot be compromised.

Drawbacks:

- *Cost:* The side chain has an initial cost, since it needs to have enough power for mining and ensuring the safety of IoT transactions.

Related tactic: Two-layer blockchain architecture (Section 3.2.12).

Example:

- *Optimized blockchain* [PS2]. A modular consortium architecture for IoT and blockchain is proposed where each chain is responsible for processing its transactions and all the side chains are attached to the main blockchain. This main chain manages access control permissions and ensures that only authorized users can have access to IoT data from one chain to another.

- *Controlchain* [PS67]. A secure architecture is presented to establish relationship attributes and access control authorization between users and devices. The blockchain database is divided into four chains: context, relationship, rules, and accountability, where all are attached to the main chain.

Variations: Plasma relies on smart contracts and Merkle Tree to arrange a hierarchical structure where numerous surrogate chains can communicate and exchange digital assets with the main blockchain [104]. Specifically, plasma implements a treelike structure that consists of child chains, parent chains, and the root chain. Overall, the plasma tactic works as follows [159]. IoT data is sent as transactions to the surrogate chains under the control of the main blockchain. If there are transactions that require a large amount of computational power, they are continuously broadcast to the main blockchain for validation. As surrogate chains are created from smart contracts, they work independently of each other and handle transactions by defining their consensus protocol and security rules. Thus, each surrogate chain monitors its transactions, eliminating the need for every blockchain node to verify all transactions performed over time on the network [89]. In addition, transactions are moved from the surrogate chains to the main blockchain when it is proven that a participant in a surrogate chain has acted maliciously.

3.2.7 IoT devices as lite blockchain nodes

Summary: Connect resource constraint IoT devices to the blockchain network through a gateway device.

Motivation: IoT mainly comprises resource constraint devices with limited computation, storage, and power that do not allow them to implement complex security protocols or process a large amount of data generated by IoT devices [89]. Thus, IoT systems can use blockchain technology to record sensor data as immutable and tamper-proof transactions. However, one of the main limitations of the blockchain is the restricted computation and data storage, since all transactions are replicated across the blockchain network [71].

Solution: Figure 3.8 shows the main components of this tactic. To facilitate the adoption of blockchain, the Bitcoin protocol identifies two types of blockchain nodes (i.e., full and lightweight nodes) [89]. The former has enough processing power and storage capacity to process transactions, mine blocks, and keep a full copy of the ledger, while the latter can only store their addresses and send transactions to the full nodes. In IoT networks, the resource-constrained IoT devices like Arduino can collect and send sensor data to resource-rich devices, as well as act as lightweight nodes due to their limited computational capabilities and battery lifetime. On the contrary, resource-rich devices can act as a gateway between the IoT network and the blockchain, as well as acting as full nodes [107]. Specifically, resource-constrained IoT devices share collected data with resource-rich devices for processing and storage. Once data is processed, they are sent to the blockchain network as a transaction using a smart contract [111].

Consequence:

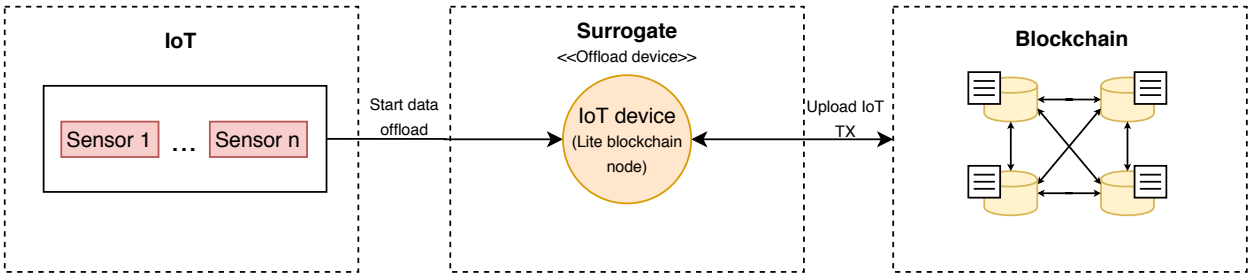

Figure 3.8: IoT device as lite blockchain node where the surrogate uploads IoT transactions to the blockchain.

Benefits:

- *Low latency:* The data offload operation in the intermediary server decreases latency, since it is in single hop proximity to IoT devices.
- *Network efficiency:* The use of Wi-Fi or short-range radio instead of broadband wireless to communicate sensors and the intermediary server reduces bandwidth consumption and improves the user experience.
- *Security:* Critical and sensitive IoT data can be processed and analyzed locally within the IoT network, which could result in better control of security and privacy levels.

Drawbacks:

- *Scalability:* The integration of resource-constrained and IoT devices with high capabilities improves the scalability of the blockchain network while ensuring performance [113].
- *Security:* The use of computationally powerful IoT devices as gateways increases security and privacy concerns about data manipulation and loss of information [89].

Related tactic: IoT device as full blockchain node (Section 3.2.8).

- *BIas* [PS23]. The system categorizes IoT devices as full, lightweight, and non-blockchain nodes according to their computation capacity and connection lifetime. In particular, the full nodes transmit the transactions from the lightweight nodes to the blockchain. The non-blockchain nodes are devices with limited capacity that cannot act as a full or lightweight client and must connect to a trusted remote node.

- *Scalable blockchain for IoT* [PS26]. The systems distinguish full, lightweight, and coordination nodes based on their power supply and hardware configuration. The full and lightweight nodes maintain constant and dynamic links in the network, respectively, and operate as nodes in the blockchain network. The coordination nodes connect devices with dynamic connections to the blockchain network.

- *A two-layer consensus* [PS40]. The system classifies IoT devices into three groups: A (server and back-end), B (edge devices and gateways), and C (low-bandwidth end devices). Devices in group A are responsible for connecting devices in group C by transmitting their transactions to the blockchain network, while devices in group B can maintain a direct connection.

- *Hybrid-IoT* [PS43]. This system categorizes IoT devices as full, lightweight and outsider nodes where full nodes participate in the consensus, and mines blocks and lightweight nodes connect to a full node to send transactions to the blockchain. Due to their limited hardware resources, the outsider nodes only sense the environment, and their data is not stored in the blockchain to prevent data overload.

- *Blockchain lightweight IoT Clients* [PS30]. IoT devices act as lightweight clients, which only store their blockchain addresses and send transactions to all nodes (i.e., the base station). The full nodes consist of a set of wireless base stations that collect transactions from the lightweight nodes.

- *Plasma* [PS32]. Plasma enables low-powered IoT devices to operate as lightweight

nodes and communicate with edge gateways that act as full nodes. Full nodes view the lightweight nodes as their clients and collect their transactions to send them to the blockchain. Plasma enables low-powered IoT devices to operate as lightweight nodes and communicate with edge gateways that act as full nodes. Full nodes view the lightweight nodes as their clients and collect their transactions to send them to the blockchain.

3.2.8 IoT devices as full blockchain nodes

Summary: Use IoT devices with high computational capabilities as full nodes to connect directly to the blockchain network.

Motivation: IoT devices with high computational capabilities connect directly to the nearest blockchain node and push transactions to the blockchain network. A full blockchain node keeps a copy of the complete blockchain and validates its transactions, as well as other transactions on the blockchain network.

Solution: Figure 3.9 shows the main components of this tactic. Powerful IoT devices, such as the Raspberry Pi, can be connected directly to the blockchain and operate as a complete blockchain node [95]. This device acts as a connector that provides communication channels and local services to resource-constrained IoT devices. Specifically, the connector assumes the role of full nodes by processing transactions and participating in the consensus protocol. To this end, the connector communicates with the nearest blockchain node through a Web3 provider and uploads IoT data to the blockchain network via a smart contract [89].

Consequences:

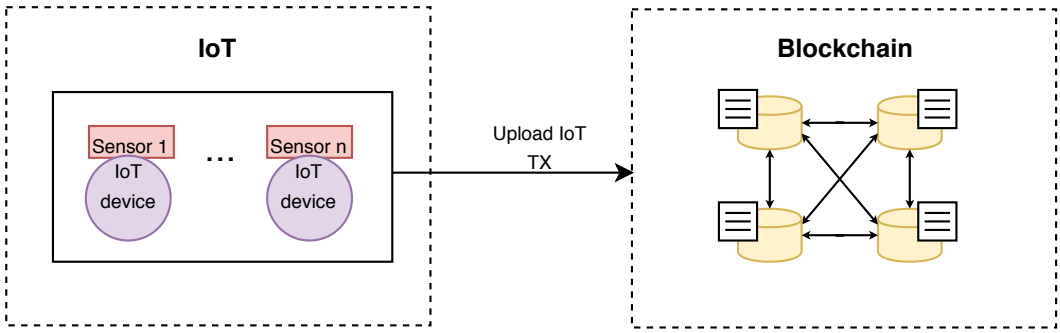

Figure 3.9: IoT devices act as full blockchain nodes.

Benefits:

- *Latency efficiency:* The deployment of a connector located in single-hop proximity of IoT devices for data processing and connecting directly to the blockchain minimizes latency in the network.
- *Bandwidth reduction:* The connector installed at the edge of the network near the IoT devices results in a lower bandwidth demand, as the data is processed locally instead of sending it to the cloud.

Constraints:

- *Lack of confidentiality:* The connector could raise security and privacy concerns about data manipulation and loss of information, as all data is available in the connector and could be manipulated and altered by malicious users.
- *Single-point-of-failure:* The deployment of a single server to process and store large volumes of data could become a single point of failure and bottleneck in the network as the number of IoT transactions increases over time.
- *Cost:* The integration of devices with strong computation capabilities to connect directly to the blockchain could increase implementation and maintenance costs.

Related tactic: IoT devices as a lite blockchain node (Section 3.2.7).

Examples:

- *Optimized blockchain* [PS2]. IoT devices with high computational resources can act as gateways in the network and transmit IoT data to the blockchain.
- *Blockchain Meets IoT* [PS75]. The system uses a management hub to connect IoT devices to Ethereum nodes through RPC calls and a JavaScript library.
- *IoT protection-blockchain* [PS14]. The system introduces intermediary servers between IoT devices and the blockchain to perform real-time processing tasks before transmitting the results to the blockchain network. Here, a publish-subscribe mechanism is used to handle computation and power-consuming resources in the blockchain network.
- *IoT data assurance* [PS18]. The control system aggregates the collected data from drones and calculates its hash before recording it on the blockchain and cloud to ensure its integrity.

3.2.9 Caching Offload

Summary: Use a cache system to offload a subset of IoT transactions processed by blockchain to make faster data operation requests.

Motivation: Due to the constraint of resources in most IoT devices, IoT systems mainly use computational and storage capabilities in the cloud [111]. However, access to sensor data in the cloud demands over a multi-hop proximity and lower bandwidth connection, which increases latency in IoT transactions and bandwidth consumption.

Description: Figure 3.10 illustrates the main components of this tactic. The caching offload tactic requires sensors running on the IoT layer, an intermediary server operating as a surrogate, and shared data storage running on the blockchain. The basic functioning of this tactic is described below [148]. The sensor data from thousands of IoT devices is sent to the intermediary server for processing and analysis before recording them as transactions in the blockchain. To perform faster data operations, the surrogate retrieves data from the blockchain and stores them locally, so that it is available to IoT devices when they need it. Therefore, access to the blockchain is only necessary when data is not available on the surrogate, which minimizes latency in the network.

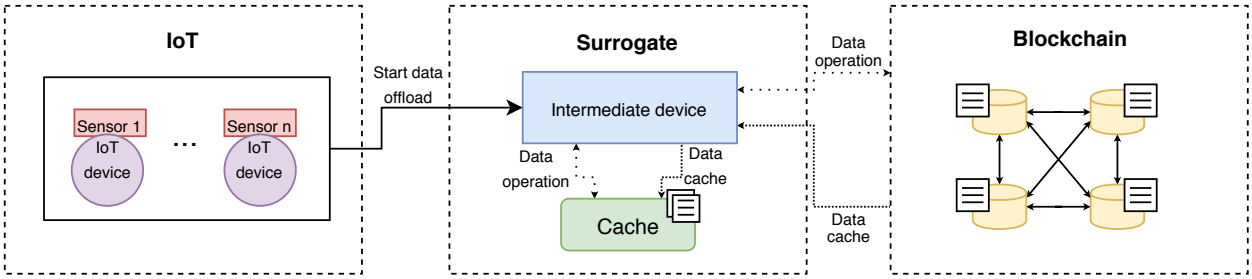

Figure 3.10: Caching offload, where the surrogate manages a cache system.

Consequences:

Benefits:

- *Improved latency:* Using caching, the retrieval of data from edge servers could be faster compared to the retrieval of all information from the blockchain. This improvement in data access also has an impact on the overall performance of the system.
- *Low throughput:* The use of caching improves the transaction throughput in the blockchain network because the edge server enables faster access to on-chain data.

Drawbacks:

- *Interoperability:* The interoperation between the caching system running on the surrogate and the blockchain nodes could be difficult and increase security concerns about data management.
- *Stale data:* The use of a cache system could lead to stale data where in each data request previously recorded data can be fetched instead of a new value of the data.

Related tactic: N/A

Example: [PS66] suggests an example of application of the caching offload tactic in Edge and Caching. Due to the limited capabilities of IoT devices, blockchain nodes rely on a caching system implemented on edge servers to reach consensus and cache resources. It ensures fast data access and improves the performance of the system.

Variation: A hardware-based caching [PS92]. This system proposes a cache technique using a Field Programmable Gate Array-based Network Interface Card (NIC) to process data requests from IoT devices before sending them to the blockchain. In particular, data requests are handled in FPGA internal memory and storage and pushed to the blockchain as transactions, reducing latency and bandwidth in the network and improving transaction confirmation.

3.2.10 Surrogate computation

Summary: Offload computation-intensive blockchain tasks (i.e., transaction processing and keeping a copy of the entire blockchain) to a surrogate to reduce computation and data

storage on-chain.

Motivation: In public blockchains, the miner and the full nodes are required to store the complete copy of the ledger and validate each transaction in order [89]. This feature improves the security of IoT systems but can also overload blockchain nodes with computation and data storage requirements due to the large amount of data sent as transactions to the blockchain [94]. Furthermore, the requirement to maintain a complete copy of IoT transactions in the blockchain nodes limits the integration of IoT devices as full blockchain nodes due to their limited resources.

Description: Figure 3.11 shows the main component of this tactic. The surrogate computation tactic requires a cloud server, a share of data storage running in the cloud, and a blockchain, respectively. This pair of components communicates to coordinate computation-intensive tasks. The basic functioning of this tactic is as follows. First, sensor data from IoT devices must be uploaded as transactions to a blockchain to enhance its immutability and integrity. Due to the limited computation and data storage in a public blockchain, it establishes connectivity to a cloud server to perform tasks that require extensive computation (i.e., hash calculation and transaction processing) [34]. Once the task is completed and the results of the data operation are sent back to a blockchain for verification [148]. If the hash is correct, then a blockchain node generates a new block and broadcasts it to all nodes in the P2P network. Each node receives the new block and validates it in consensus before adding it to the end of the chain.

Consequences:

Benefits:

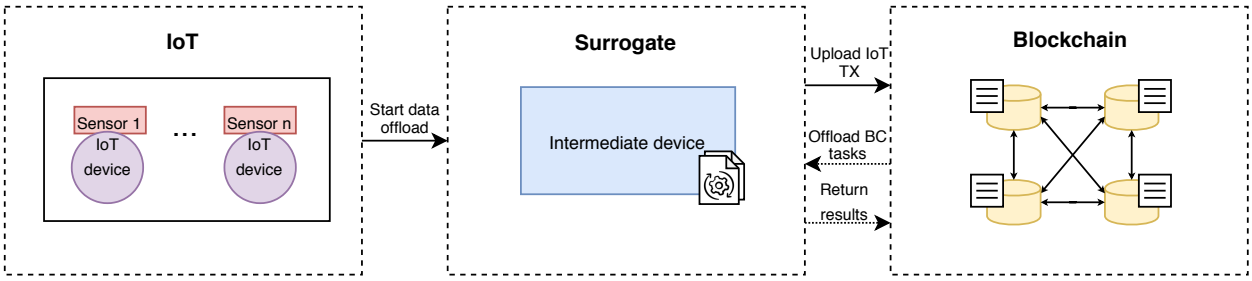

Figure 3.11: Surrogate computation, where the surrogate processes blockchain tasks.

- *Computation efficiency:* The use of a cloud server as a surrogate helps to reduce computation and data storage loads on blockchain nodes and reduce latency in transaction confirmation.

Drawbacks:

- *Data immutability:* Since blockchain connects to a cloud server to process data and perform computation-intensive tasks, IoT data could be altered and manipulated by cloud service providers.

Related tactic: N/A

Example: The systems that implement the surrogate tactic maintain a list of edge servers that are allowed to connect to the blockchain and call smart contract functions. For example, Edge and Caching [PS66] is based on edge servers to maintain a P2P network and execute computationally expensive tasks such as hashing. Specifically, the authorized edge servers in the smart contract receive all the required information to calculate the hash and return the output to a blockchain network for verification. This verification process consists of a proof of execution on-chain.

3.2.11 Sharding

Summary: The majority of IoT systems have real-time data sharing requirements that require improvements in the transaction confirmation time on the blockchain. The use of sharding increases transactional throughput in blockchain networks with minimal disruption to IoT users.

Motivation: The large amount of data generated by IoT devices results in a large number of transactions that must be uploaded to a blockchain. However, in a public blockchain, the miner nodes are responsible for keeping a complete copy of the ledger and validating every transaction in order. Since the blockchain cannot process more transactions than the capacity of a single node, it could become a bottleneck in the case of a high number of transactions.

Description: Figure 3.12 shows the main components of this tactic. The sharding tactic consists of spreading the computation and data load across the blockchain network to reduce the transaction confirmation time [89]. This means that a subset of mining nodes processes a subset of transactions generated by IoT devices instead of processing all transactions on the blockchain network. Each node is only responsible for maintaining information related to its partition (i.e., shard) and maintaining its transaction history [85]. The subset of the transaction consists of a header (i.e., identifier) and a body (i.e., all transactions belonging to a specific group). Once a transaction is verified within a shard, the entire shard is updated to ensure that all nodes within the shard have the same information. Additionally, a shard can trigger events on other shards to exchange digital assets, which is known as cross-shard communication. These arrangements ensure that multiple transactions can be processed simultaneously and enhance security in the blockchain network.

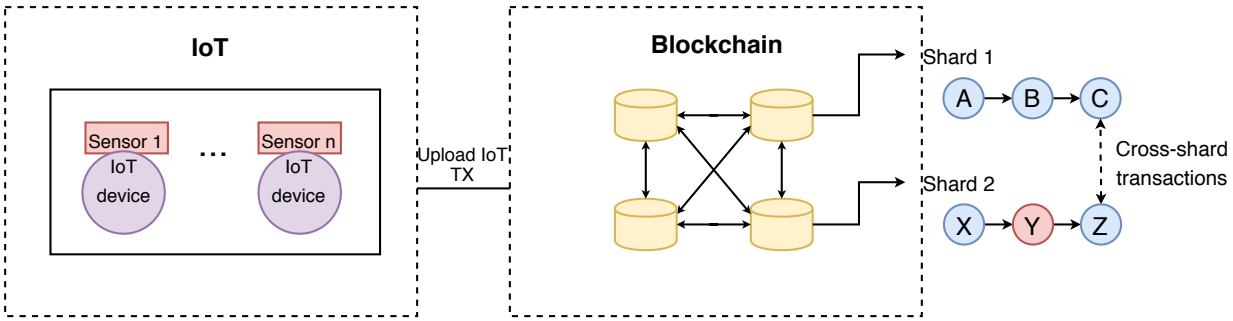

Figure 3.12: Sharding where the surrogate handles shards.

Consequences:

Benefits:

- *Faster transactions:* Since one of the main advantages of sharding is the ability to process transactions in parallel, it can process 10 times the number of transactions performed by traditional blockchains per second.
- *Low data storage and cost:* The use of shards facilitates the storage of a large amount of IoT data as transactions at low cost because each blockchain node handles a small portion of the data or keeps a complete copy of the ledger.

Drawbacks:

- *Data sharing:* The sharding enables transaction exchanges between shards, but cross-shard communication is still challenging. When a specific participant in one shard requires information that is not within its shard, it has to identify which shards contain the required information and exchange it for transaction processing.

Related tactic: N/A

Example: An example of the sharding tactic has been identified in [PS42]. The system consists of multiple micro-blockchains, also known as shards, where each of them is responsible for receiving transactions, broadcasting them to the P2P network, and making the consensus. Each shard and the main shard run the PBFT consensus protocol twice to reach a consensus and create the blocks. The main shard consists of some important nodes in each shard that are responsible for making the final consensus and generating the blocks to be attached to the main chain.

3.2.12 Two-layer blockchain architecture

Summary: Enhance the scalability of blockchain by relying on a two-layer blockchain architecture.

Motivation: With thousands of sensors collecting data from the environment, a large number of transactions must be uploaded to the blockchain [89]. Since sensor data is replicated in all blockchain nodes, it affects the blockchain size and influences the consensus protocol. Furthermore, the high storage requirements of blockchain systems put more limitations on the integration of resource-constrained IoT devices such as blockchain nodes [111].

Description: Figure 3.13 shows the main components of this tactic, including IoT devices, multiple public blockchains and a private blockchain. The two-layer consensus tactic can enable interoperability between public and private blockchains and solve scalability issues, as well as increase throughput in public blockchains. The basic functioning of this tactic is described as follows [113]. Firstly, resource-rich IoT devices send transactions to public blockchains in the first layer, where each of the chains calculates a hash from a group of transactions. After the hash calculation, the public blockchains send them periodically

to a private blockchain in the second layer for verification and authentication. Once the transactions have been verified, they are sent back to the corresponding public blockchain for immutable storage.

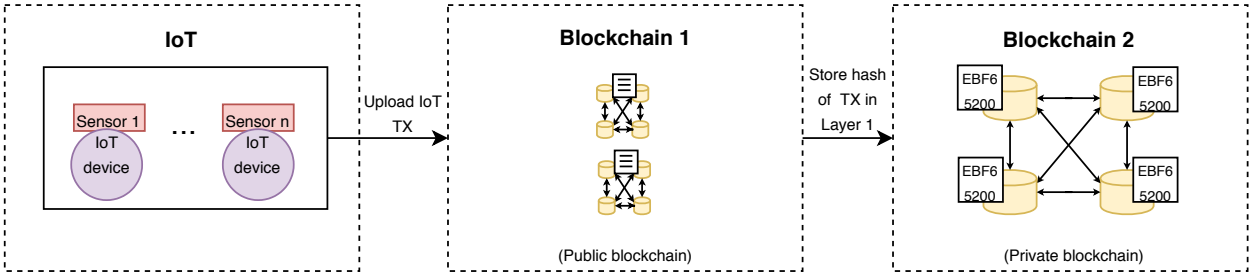

Figure 3.13: Two-layer consensus, which supports a public and private blockchain.

Consequences:

Benefits:

- *Scalability:* The use of multiple public blockchains and a private blockchain improves the scalability of the blockchain network, since IoT transactions are recorded in a distributed database. In addition, the use of offline data storage can alleviate the storage requirements in the blockchain nodes in the second layer.
- *Integrity:* Since the hash of IoT transactions processed in the first layer is periodically recorded in the second layer, it makes it easy to detect data forgery.

Drawbacks:

- *Limitation in private blockchain:* Using a private blockchain, the maximum number of validators tested in previous work is 20 [89], which makes it difficult to integrate thousands of powerful IoT devices as blockchain nodes.

Related tactic: Off-chain data storage (Section 3.2.5).

Example:

- *Hybrid-IoT* [PS43]. A PoW-based sub-blockchain is created to achieve distributed consensus among IoT devices as nodes of the blockchain network. Each sub-blockchain consists of a group of IoT devices that follows a set of rules called sweet-spot guidelines to define in which way IoT devices can establish a sub-blockchain. All sub-blockchains are connected to a PBFT interconnector framework that handles the interoperability among multiple sub-blockchains.

- *Two-layer consensus* [PS40]. The base-layer and top-layer are deployed, and different class nodes are defined based on their capabilities and connection lifetime. On the one hand, the base-layer consists of class B and C nodes and considers a hybrid consensus protocol (i.e., PoW and PoS) to improve the scalability and transaction time of the blockchain network. In contrast, the top-layer is formed by class A nodes selected by managers and ensures that base-layer blocks are performed in a randomly and transparent manner by following a non-byzantine fault tolerance algorithm.

3.3 Discussion

We use the design science-based evaluation approach for patterns proposed by Petter et al. [103] and adapt it to reflect on architectural tactics for blockchain-based IoT systems. In this chapter, we view tactics as a method for moving from ad-hoc design to a systematic procedure that investigates how design decisions can affect the quality attribute requirements. Taking this qualitative approach, we would not only eliminate guess, but also provide informed

decisions and a set of architectural options to satisfy particular system qualities. Table 3.2 presents the evaluation criteria tailored to the context of tactics.

Table 3.2: Architectural tactics for IoT systems supported by blockchain.

Evaluation criteria	**Traditional definition**	**Tactic definition**
Plausible	The degree to which a concept is more than just a belief [67, 127].	Tactics are design decisions that are made in a particular context or environments to achieve specific quality attribute requirements [6].
Effective	The degree to which a concept describes the phenomenon under study parsimoniously and stimulates inquiry [67, 112, 129].	Tactics are described in understandable language; reason about the problem, its causes, and provide a possible solution.
Feasible	The degree to which a concept is operationalizable or workable [67].	Tactics can be implemented as described and applied based on the context.
Predictive	The degree to which a concept is capable of predicting outcomes for given conditions [67, 127, 129, 136].	Tactics influence in the achievement of particular system qualities.
Reliable	The degree to which different researchers certifiable a concept using different methods [19, 67, 130, 129].	Tactics can be reused in system development to achieve desired system qualities, and can produce similar results regardless of the implementer or technique.

Plausible: The tactics were derived from the results of the systematic review of the literature (SLR) to identify the common architectural design decision in blockchain-based IoT systems reported in Chapter 2. It includes analysis of the state-of-the-art and practical applications on the integration of blockchain and IoT to select the primary studies using a categorization of architecture decisions. Decisions were related to the distribution of storage and computation, blockchain scope, consensus protocol, blockchain data structure, and blockchain deployment. The SLR results showed that design decisions have led to the identification of 13 tactics to satisfy particular quality attribute requirements, such as performance, scalability, security, and interoperability.

Effective: The tactics were described in a systematic and comprehensible way using the template presented in [73]. The template includes a meaningful motivation, a brief description, the constraints, one or more examples, the dependencies (optional), and a list of variations. Our interest in using this template is to guide architects and developers in the process of designing blockchain-based IoT systems to achieve the desired qualities. Moreover, the description of each of the components in the tactic can help architects to extend their reasoning towards the effect of their decisions. As a result, the implementation of the tactics will indicate how easily and accurately architects understand the information conveyed in the tactic description.

Feasible: The tactics described in this chapter are feasible and used in particular contexts to achieve the desired quality attribute responses. A key consideration in assessing the feasibility of each tactic is the identification and analysis of the constraints in the implementation. These constraints might render a tactic impractical for some specific contexts

and valuable in others that ensure successful implementation. For example, in this chapter, the tactics presented were extracted from primary studies and implemented in the implementation phase of small-scale prototypes and real case studies to observe their feasibility. Therefore, tactics seem to be operationalizable within the specific contexts of their intended use.

Predictive: In this chapter, we presented a catalog of 12 architectural tactics for blockchain-based IoT systems to achieve requirements for particular quality attributes such as performance, scalability, security, and interoperability. Since they were derived from the SLR results on architectural decisions presented in Chapter 2, they have been assessed in primary studies to determine whether they satisfy the expected qualities of the system. For example, *encryption on-chain data tactic* was designed to improve the security of IoT data by encrypting the data before recording and replicating it in blockchain nodes. Moreover, the effect of the application of this tactic could vary depending on the context, and should satisfy the same system quality.

Reliable: The tactics are design decisions that can be reused in the development of blockchain-based IoT systems to satisfy the desired quality attribute requirements. On the one hand, independently of the application context, the tactic should produce consistent results. However, the use of a qualitative or quantitative assessment approach should not affect the results. For example, *off-chain storage tactic* uses a third party offline data storage to record IoT data while keeping the digital hash of critical data on-chain. Regardless of the off-chain storage (e.g., cloud, fog, or local database), the tactic improves the scalability of the system since IoT data can be offloaded to an external storage.

3.4 Conclusion

This chapter presented a catalog of architectural tactics derived from common design decisions identified as a result of the SLR described in Chapter 2. We selected the common design decisions from the primary studies and codified them into reusable tactics for blockchain-based IoT systems. A tactic can serve as a guide or reference for architects and developers in the system design process to satisfy the qualities of a particular system [119, 131]. In particular, we delivered a catalog of 13 tactics, which were grouped based on the quality attribute requirements that they satisfy in the system, such as security, scalability, performance, and interoperability. Although we identified a tactic for interoperability, it is a key quality of the system to guarantee the successful operation of this category of systems, especially when IoT devices from different vendors interact in the environment. We highlight that the identified tactics are not exhaustive; instead, there are other system qualities that are not considered in primary studies, such as adaptability and mobility, which are relevant in the development of this category of systems. In general, the catalog can help architects and developers identify tactics that can meet the quality requirements of the system.

Furthermore, we identified gaps and opportunities for our work in the remainder of this thesis, as follows: (i) evaluate the real-world impact of identified architectural tactics in existing architectures for IoT systems supported by blockchain and (ii) explore the trade-offs between the quality attributes and identified tactics. These opportunities for research require intensive collaboration between academia and industry, considering the fact that meaningful IoT systems consist of thousands of devices that collect a large amount of data that need to be processed and analyzed.

Meanwhile, the next chapter presents a set of reference architecture styles and variants that serve as underlying architectures that can drive the development of IoT systems supported by blockchain. We evaluate styles using ATAM to assess how they can affect the

achievement of the quality attribute requirements of this category of systems. Analysis of the results leads to the identification of some variants, which can provide some guidelines for the development of this category of systems.

Chapter Four

Reference Architecture Styles for IoT systems supported by blockchain

In chapter 3, we derive the commonly identified design decisions in the primary studies and codify them as architectural tactics that can be used in the development of IoT systems supported by the blockchain to achieve particular quality attributes. In this chapter, we provide a set of reference architecture styles and variants that can guide software architects and developers in the design of software architectures for this category of systems (thereby addressing RQ3). In particular, Architectural Tradeoff Method Analysis (ATAM) is used to understand the tradeoffs of existing styles and assess their fitness with regard to particular system qualities. The results of the ATAM analysis have led to refined architectures. We document styles and variants using the architectural pattern language to facilitate their adoption when designing this category of systems. Finally, a quantitative analysis of styles and variants is performed through simulation to complement the ATAM analysis and evaluate its applicability and effectiveness in terms of latency and network usage.

4.1 Overview

An *architectural style* determines "the vocabulary of components and connectors that can be used in instances of these styles and a set of constraints on how they can be combined" [42]. In simple terms, a style comprises a set of design decisions that are applied to a specific context to satisfy the desired quality attributes and tradeoffs [126]. In practice, one or more architectural styles can be used for the architectural design of a software system that effectively satisfies the qualities of the particular system. However, one of the main issues in designing IoT systems supported by blockchain is the data-driven nature of IoT, characterized by high velocity, high volume of data, and high mobility, making data security a challenge. On the other hand, blockchain presents technical constraints of a complex nature that can limit its adoption in IoT systems at scale, such as limited space, immutability, and excessive computational power, among others.

Several approaches have been investigated for the adoption of blockchain in IoT applications [146, 157, 30, 123, 56, 28, 74], but only a few attempts have been made to architect IoT systems supported by blockchain [111, 147]. Although these attempts usually propose some architectural alternatives that meet the desired quality attributes, they do not investigate the design decisions and constraints imposed by the IoT and blockchain. Thus, there is a general absence of systematic investigations on architectural styles and architectural evaluation approaches that can be used to understand the tradeoffs inherent in architectures, inform design refinements, and decide on architectural choices that effectively realize the desired quality attributes in the IoT system supported by blockchain.

This chapter codifies a set of reference architecture styles and variants for on- and off-chaining data and leverages Architecture Tradeoff Analysis Method (ATAM) and simulations to guide the architectural design of this category of systems. The reference architecture styles can serve as generic model solutions for given problems, considering the characteristic of

the environment [41]. They are generic in nature, but they can have variants to address specific concerns. Both styles and variants can serve as guidelines for software architects and developers to advance the design of software architectures for this category of systems that meet the desired quality attributes. In particular, we review a set of reference architecture styles by inspecting representative examples from the literature. We use these styles to reason about the architectural design decisions that are made in a particular context to meet all the desired quality attributes. The key decisions identified in existing architectural styles include on-chain/off-chain storage, type of blockchain, and consensus mechanism to satisfy system qualities such as performance, while trade-offs between quality attributes such as performance, availability, and security.

We use the Architecture Tradeoff Analysis Method (ATAM) [65], one of the widely and widely used techniques for understanding the tradeoffs implicit in software system architectures, to systematically assess the general fitness of the existing reference architecture styles for IoT systems supported by the blockchain in relation to the desired characteristics of the system and tradeoffs of the system. This evaluation comes at the price of improving some quality attributes and affecting others, which leads us to reason about the architectural design decisions that influence the achievement of particular quality attributes. In particular, the evaluation of reference architecture styles generates outputs, such as sensitivity, tradeoffs, and risk points, which have led to refinements of the architectures. These refinements are known as variants in this chapter and correspond to small changes in the basic architecture to satisfy the quality attributes and tradeoff points of the IoT systems supported by the blockchain.

To apply the ATAM procedure, we first collect scenarios from a healthcare case study that clearly state specific quality attribute requirements, annotated with stimuli and responses. Third, we evaluate scenarios against the qualities of the system, resulting in risks, sensitivity, tradeoffs, and risk points. This analysis has led to refinements of the reference

architecture styles, defined as variants that meet particular characteristics of the system. Both styles and variants are documented using the architectural pattern language suggested in [73] to facilitate their adoption in IoT systems supported by blockchain, including introduction, motivation, description, constraints, and examples. To provide concrete results of tradeoff points between quality attribute requirements and design decisions, we complement the ATAM analysis with a quantitative evaluation using simulation. In particular, we evaluate the reference architecture styles and variants in terms of latency, network usage, energy consumption, and cost to get strong evidence of the suitability of the architecture styles for IoT systems supported by blockchain.

- A set of reference architecture styles and variants for IoT systems supported by blockchain derived from the primary studies on the integration of blockchain and IoT to guide software architects in the development of this category of systems to achieve particular quality attributes and tradeoffs.
- A qualitative evaluation using ATAM to assess the general fitness of the reference architecture styles for IoT systems supported by blockchain with respect to quality attributes. The results of the ATAM analysis have led to refinements of the architectures, resulting in three variants.
- A quantitative evaluation of the styles and variants for IoT systems supported by blockchain using simulation to demonstrate its applicability and effectiveness in terms of latency, network usage, and energy consumption.

The remainder of this chapter is organized as follows. Section 4.2 describes the architectural evaluation methods. Section 4.3 introduces a motivation example and quality goals. Section 4.4 describes the candidate styles. Section 4.5 presents the qualitative and quantitative evaluation along with the results of the analysis. Section 4.6 discusses the main findings

of our study and the threats to validity. Finally, Section 4.7 presents previous studies and Section 4.8 concludes the chapter.

4.2 Architectural evaluation methods

Most architectural evaluation approaches are based on scenarios [64, 100, 13, 65, 29] and use quantitative evaluations to determine the satisfaction of quality attribute requirements. In general, these approaches generate scenarios and assess candidate architectures in terms of the quality attributes of interest.

- *Software Architecture Analysis Method (SAAM) [64]:* Is the first well-known scenario-based software architecture method that translates quality attributes into scenarios to assess how well the candidate architecture meets them. In addition, SAAM assesses quality aspects in software architectures in terms of weak or strong points to rank between them. Even though SAAM was designed to analyze modifiability in a software architecture, it is used for testing general non-functional requirements.

- *Cost-Benefit analysis method (CBAM) [100]:* Is an architectural evaluation method that supports software architects in making decisions to maximize their gains, meet their requirements, and minimize their risks. In contrast to the technical tradeoff analysis performed in ATAM, CBAM considers the cost and benefit of each architectural decision to achieve the qualities of the system. This analysis can also be used to justify the selection of a candidate architecture that meets particular quality attributes.

- *Architecture-Level Modifiability Analysis (ALMA) [13]:* Is a scenario-based software architecture method that focuses mainly on modifiability and makes explicit assumptions to achieve this particular quality attribute. ALMA consists of five steps, namely,

goal selection, description of the software architecture, elicitation of scenarios, changes in scenario evaluation, and interpretation. Unlike ATAM, ALMA does not perform a tradeoff analysis.

- *Architectural Trade-off Analysis Method (ATAM) [65]:* One of the widely and commonly used techniques to evaluate architectural choices and decisions in light of the quality attribute requirements. ATAM provides us with a systematic procedure to understand the potential tradeoffs and constraints implicit on the candidate architectures and evaluate how well they address the quality attribute requirements. Although ATAM has evolved from SAAM, it uses scenarios to assess potential architectures and mitigate the risks in them at the early stage of the software development life cycle. The ATAM procedure requires skateholders to identify goals, constraints, system functionality, and desired quality attributes to create scenarios and evaluate them against the quality attributes. The output of this analysis leads to the identification of trade-offs, sensitivity points, and risks in the architecture.

- *Family – Architecture Analysis Method (FAAM) [29]:* It is a systematic evaluation assessment method for information systems that focuses on the quality attributes of interoperability and extensibility. FAAM enables skateholders to identify and express future changes and cases of the system for exploration and analysis.

Despite the variety of architectural evaluation methods in the software engineering community, the selection of ATAM was guided by (i) the fact that it has been recognized as an effective scenario-based method for architectural analysis and (ii) its popularity among software architectural projects ranging from automotive to mission-critical systems such as Battlefield Control System (BCS) and EOSDIS Core System (ECS) [63, 65, 24].

4.3 Motivation example and requirements

We use a healthcare case study to generate scenarios that clearly state particular quality attributes of IoT systems supported by blockchain in terms of stimuli and response.

4.3.1 Sleep apnea example

We adopt the sleep apnea case study provided in [138] to demonstrate the importance of systematic analysis and architectural modeling of styles and their variants. Sleep apnea is a serious sleep disorder that occurs when breathing is interrupted for 10 seconds or even longer periods while sleeping [33]. According to the World Health Organization, millions of people around the world suffer from Obstructive Sleep Apnea (OSA) due to their sedentary lifestyles, unhealthy diets, and increasing life expectancy [124]. In more severe manifestations, sleep apnea increases the risk of high blood pressure, stroke, and even heart attack. To determine the level of sleep apnea, the heartbeat rate and blood oxygen level (SPO2) are used to calculate the Apnea Hypopnea Index (AHI) as follows:

- *Mild:* $5 \leq \text{AHI} < 15$ per hour
- *Moderate:* $15 \leq \text{AHI} < 30$ per hour
- *Severe:* $\text{AHI} \geq 30$ per hour

Assume that Alice, a 60-year-old woman, suffers from sleep apnea. She uses a pulse oximeter (e.g., a portable IoT device) to monitor her heartbeat rate and blood oxygen level (SPO2) during sleep. The IoT device transmits the sensor readings to Alice's mobile phone, which acts as a gateway device and forwards the data to the cloud for processing and analysis. One night, Alice has consecutive episodes of low blood oxygen that provokes her unconscious.

Her phone triggers a real-time alarm to the closest Emergency Hospital Staff (EMS), doctor, and family. The EMS accesses to Alice's health data from her Electronic Health Record (EHR) in the cloud which is shared with other healthcare providers (i.e., hospitals, laboratories, pharmacies, and healthcare insurances) to enable automatic services and ensure fast intervention. However, the EHR also includes sensitive and critical information (i.e., home address, email, security number, etc.) that could be manipulated and altered by unauthorized health care providers in the cloud. The general usage scenario for the healthcare remote monitoring system is illustrated in Figure 4.1.

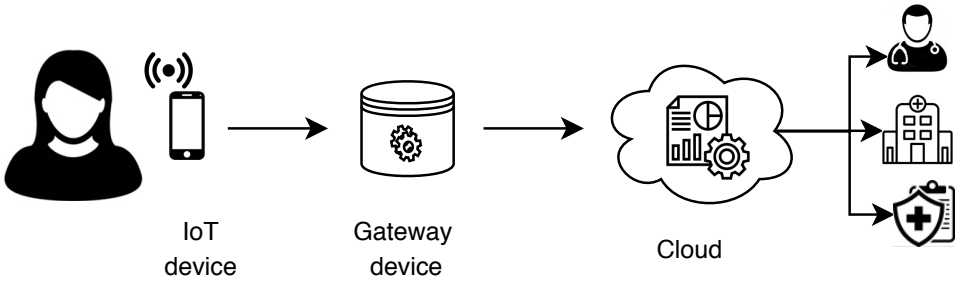

Figure 4.1: A remote health monitoring example.

From the above scenario, we obtain a set of high-priority requirements as follows.

- *Requirement 1 (R1):* The EHR could include sensitive and critical information (e.g., home address, security number, diagnosis) that must be recorded in an immutable and verifiable way and securely shared between healthcare providers.
- *Requirement 2 (R2):* The health data shall be processed close to where it is collected to ensure quick analytic data and minimize latency and bandwidth consumption in the network.
- *Requirement 3 (R3):* The health system must guarantee efficient use of hardware and software resources to reduce energy consumption.

Overall, the requirement *R1* has security implications that should be interpreted as the common quality attribute requirements that need to be considered in the design of IoT systems supported by the blockchain, as described in Section 2.3.1. Furthermore, the requirements *R2* and *R3* should be translated into performance and scalability goals to be satisfied for the development of this category of systems.

4.4 Reference architectural styles

We identify three reference architecture styles for IoT systems supported by blockchain, derived from the primary studies on the integration of blockchain in IoT described in the SLR. The styles are (i) directly connected IoT-blockchain, (ii) indirect connected IoT-blockchain, and (iii) hybrid connected IoT-blockchain, and have three variants. These variants are potential extensions of the styles driven by some scenarios and characteristics that have informed their inception and refinement. It should be noted that this study does not include a full list of conclusive architectures for this category of systems. Instead, the three selected architectural styles capture the core features of IoT and blockchain technologies and intersect many IoT applications that can benefit from the blockchain as a distributed ledger. Similarly to styles, the variants presented are also based on the core essence of IoT and blockchain, limitations, and design issues.

In general, the styles and variants presented can provide software architects and designers with blueprints and guidelines to support their current and new developments in the architecture of this category of systems. We use the architectural pattern language provided in [73] to describe the reference architecture styles and variants in a concrete and systematic manner to ease their adoption in the domain, as follows:

- *Introduction:* Brief explanation of how styles and their variants work.

- *Motivation:* Rationale behind the implementation of the styles and their variants using the industrial case study presented in Section 4.3.
- *Description:* Detailed explanation of the components in the styles and their variants.
- *Constraints:* Conditions to consider when implementing styles and their variants.
- *Related styles:* Relation with other styles.
- *Example:* Application of styles and their variants to existing IoT systems supported by the blockchain.

4.4.1 Architectural Style I: Directly connected IoT-Blockchain

- *Introduction:* IoT networks comprise resource-constraint devices with low memory, storage and battery power and resource-rich devices with powerful resources to perform communication or computationally expensive tasks (i.e., mining and transaction processing). A directly connected IoT-Blockchain architecture design enables the integration of resource-rich IoT devices as full or light blockchain nodes to deploy, process, and store transaction data.
- *Motivation:* A scenario for directly connected IoT-Blockchain architecture design is the following: While Alice sleeps, the oximeter in her finger has reported low blood oxygen to her mobile phone. Immediately, it sends an alarm to the health system in the cloud to update Alice's health records. However, the transfer of a large amount of data to the cloud can lead to high latency and bandwidth consumption and response time delay. Furthermore, health records can be easily leaked, modified, or manipulated by unauthorized providers in the cloud (i.e., doctors, laboratories, insurance companies, etc.). To address these issues, health data can be processed and stored in Alice's phone, acting as a blockchain node.

- *Description:* Figure 4.2 shows the main components of the design of the directly connected architecture where resource-rich IoT devices (i.e., vehicles, surveillance cameras, etc.) can join the blockchain network as full or light nodes without the need of intermediaries. As full blockchain nodes, IoT devices can hold a copy of the entire ledger, while light nodes can only download a portion of the blockchain (i.e., the block headers) to validate the authenticity of the transactions. Specifically, IoT devices collect data from the environment and extract its value through a web or mobile application. This web or mobile application connected to a blockchain node uploads the extracted value as a transaction to the blockchain network through a smart contract. Thus, the web or mobile application serves as the communication interface between the IoT devices and the blockchain. However, in this approach, resource-rich IoT devices can be exhausted with computation (i.e., transaction processing) and data storage loads (i.e., keeping a copy of block data). Therefore, a possible solution to increase resources in IoT devices and accelerate blockchain adoption can be to offload blockchain-related tasks to centralized architectures (i.e., the cloud). That is, cloud resources can be used to handle transaction processing and to store a copy of the entire blockchain ledger.

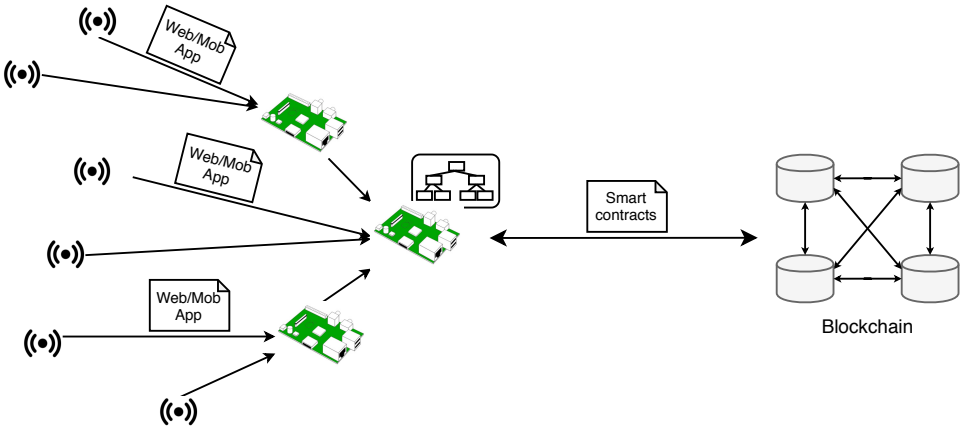

Figure 4.2: Directly connected IoT-Blockchain.

- *Constraints:* This architecture design assumes that resource-rich IoT devices have

enough computing resources and battery lifetime to operate as full or light blockchain nodes. Furthermore, this design may not benefit real-time IoT applications since blockchain-related tasks are offloaded to the cloud, which has some disadvantages such as high latency and bandwidth consumption, high maintenance cost, and security issues.

- *Example:*
  - In [79], common architectural design issues for the development of blockchain-driven IoT services are investigated, including the location of blockchain endpoints, the distribution of business logic and data, and mechanisms for cyber-physical integration. This further presents four architectural styles for the same category of systems called fully centralized, pseudo-distributed things, distributed things, and fully distributed.
  - In [89], two scenarios are presented for integrating IoT devices with the blockchain, where IoT devices can work as traditional IoT nodes or as blockchain nodes based on their computing power and battery life. For scenario 1, resource-rich IoT devices operate as full/lite blockchain nodes to send transactions to the blockchain. For scenario 2, a gateway device is deployed between resource-constrained IoT devices and the blockchain to push transactions to the blockchain.
  - In [147], two approaches for blockchain-enabled IoT systems, namely tight and loose integration, are investigated. The former enables devices with limited resources to connect to a resource-rich IoT manager to send data transactions to the blockchain. The latter relies on IoT devices with high computational resources to deploy blockchain functionalities. In addition, both approaches use an external cloud to offload blockchain data and tasks.

Although these studies suggest the use of IoT devices as full or lite blockchain nodes and

handle blockchain functionalities (i.e., transaction processing and mining), they do not perform any evaluation to demonstrate the applicability of the proposed architectures.

- *Variation:* **Offload blockchain tasks to the fog.** The architectural style assumes that blockchain-related tasks (i.e., transaction processing and storing a copy of global block data) can be offloaded to the fog rather than to the cloud to minimize latency and bandwidth consumption in the network. In this case, fog nodes located in a single-hop proximity of IoT devices can be used to improve the performance of the blockchain and improve security in the IoT system [148].

4.4.2 Architectural Style II: Indirectly connected IoT-Blockchain

- *Introduction:* The majority of IoT devices (i.e., sensors and RFID tags) still do not have enough computational resources and battery life to handle blockchain-related tasks [89]. A possible solution consists of implementing blockchain in dedicated blockchain nodes and some resource-rich IoT devices (i.e., gateways, routers, and management hubs, etc.) that can serve as a controller between resource-constraint IoT devices and the blockchain.

- *Motivation :* A scenario for indirectly connected IoT-Blockchain architecture design is the following: Alice's uses her mobile phone to locally process heartbeat and blood oxygen readings from the oximeter device to reduce the amount of data sent to the cloud. However, with the increasing amount of sensor readings collected over time, her mobile phone could be burdened with computation and data storage loads. To solve this issue, a gateway device could be deployed between Alice's phone and the blockchain to ensure rapid analytic and minimize latency in the network.

- *Description:* Figure 4.3 shows the main components of the indirect connected architecture design that differs from style I in the deployment of a gateway in a single-hop

proximity to the IoT devices. Specifically, resource-constrained IoT devices with limited memory, storage, and battery lifetime sense the environment and transmit sensor readings to a gateway with powerful resources through wireless communication. The gateway then extracts value from sensor readings through a web or mobile application, which also connects to a blockchain node to upload transaction data to the blockchain via a smart contract. However, the gateway can be overloaded with data storage and computation tasks due to the large amount of data generated by IoT devices over time [79]. Thus, a possible solution to increase the computation at the edge could be the implementation of multiple gateways to handle the computation and data storage loads generated in IoT systems.

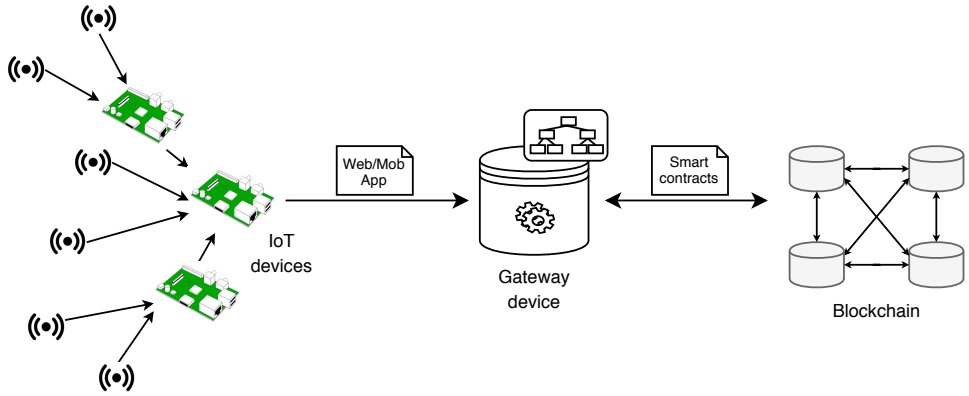

Figure 4.3: Indirectly connected IoT-Blockchain with a single gateway as a controller.

- *Constraints:* This architectural style assumes that the gateway is always available and already exists as a controller between resource-constraint IoT devices and the blockchain. Although its deployment can reduce latency and bandwidth consumption and improve security in the IoT network, it can be vulnerable to attacks that result in data theft and data forgery [89].

- *Example:*

  – In [30], a lightweight blockchain-based architecture is presented for IoT that is

managed centrally to reduce energy consumption. The architecture includes three tiers, called the smart home, the overlay network, and cloud storage, which exchange transactions with each other. In particular, the smart home consists of IoT devices, local IL and a local storage, where the local IL works as a private blockchain within the smart home. The overlay network relies on high-computing power devices to create a distributed network and process transactions from the smart home. Finally, cloud storage groups transactions into blocks based on the smart home.

- In [128], a distributed control system based on blockchain is proposed for edge computing that consists of devices, a mesh of edge nodes, and cloud services. First, physical devices are emulated as elements of the automation process, and the collected data is sent to the edge network for processing, storage, and network services. Finally, the edge nodes submit data as transactions to the blockchain provided as cloud services using smart contracts.

The examples presented in this section provide important information on the use of a middle layer of resource-rich devices (i.e., gateways or edge nodes) to connect resource-constrained IoT devices to the blockchain. However, it will be greatly improved if an evaluation in terms of performance, such as latency, network usage, and energy consumption, can be considered.

- *Variation:* **Multiple resource-rich IoT devices as controllers.** The style implements two or more gateways to analyze IoT data at the edge of the network. The gateways follow a master and slave model, where the master collects sensor readings from resource-constrained IoT devices and distributes them among slave gateways that process IoT data and forward the results to the master. Using a web or mobile application, the master gateway connects to the blockchain network to upload sensor reading values using a smart contract [122].

4.4.3 Architectural Style III: Hybrid connected IoT-Blockchain

- *Introduction:* With the increasing number of IoT systems, a large amount of data and IoT devices need to be managed in a secure and transparent manner to meet business requirements. A hybrid connected IoT-Blockchain style implements multiple gateways and blockchains to improve data management and enable separation of concerns among different types of transactions.

- *Motivation:* A scenario for hybrid connected IoT-Blockchain is the following: Multiple healthcare providers need access to Alice's EHR to ensure fast intervention and automate health services due to her severe apnea condition. However, data access requirements regarding who and what portion of the data can be shared can differ from one provider to another. For instance, the HMS needs to have access to personal details (i.e., home address, security health number), while the insurance provider only needs to know Alice's current status to facilitate insurance services. Therefore, two or more blockchains can be implemented to enable the separation of concerns among healthcare providers and with different requirements.

- *Description:* Figure 4.4 shows the main components of the hybrid connected IoT-blockchain style that differs from style II in the number of gateways and blockchain networks implemented to improve performance and improve security of IoT systems. Specifically, resources-constrained devices sense the environment and communicate with gateways at the edge of the network for data processing and analysis. The gateways follow a master and slave model, where the master collects sensor readings from resource-constrained IoT devices and distributes them among slave gateways to calculate the root hash of the Merkle tree. The slaves then forward the result to the master gateway that connects to a blockchain node to push the hash as a transaction to the corresponding blockchain using a smart contract.

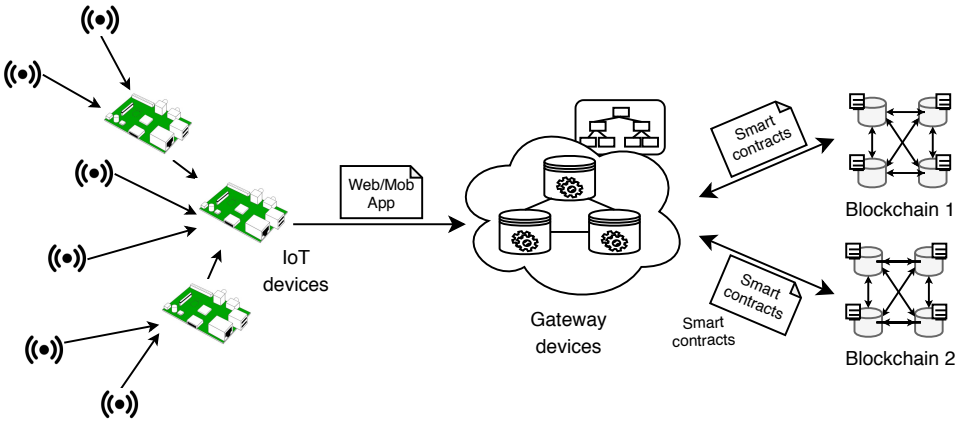

Figure 4.4: Distributed IoT-blockchain style with two gateways and specialized blockchains.

- *Constraints:* This architectural style assumes the deployment of multiple gateways and specialized blockchains to improve response time and allow separation of concerns between different types of transactions. However, it could lead to a high cost on premise-equipment and maintenance since many gateways need to be implemented based on data generated by IoT devices. Furthermore, the deployment of two or more blockchain networks could increase energy consumption throughout the network, as blockchain networks require a huge amount of power to perform mining.

- *Example:*

  – In [57], a cross-chain framework is proposed for secure and efficient IoT data management that integrates multiple blockchains. The framework is based on a consortium of blockchain, side chains, and edge smart devices. The consortium blockchain acts as a control station, while the side chains operate as the backbone blockchain for specific IoT scenarios. In particular, IoT transactions are sent to a particular side chain, which creates off-chain channels to connect to the consortium blockchain using a notary mechanism.

  – In [60], a multiple blockchain architecture is presented to exchange information between blockchain systems that consists of three layers, namely basic, blockchain,

and multichain communication. The basic layer includes the network, storage, sanbox, and database modules for system operation. The blockchain layer implements the data structure, consensus mechanism, and encryption, and the multichain layer handles transaction confirmation and assets.

  - In [58], a novel architecture is introduced to enable interoperability between blockchains, that is, the source and destination chain. The source chain selects nodes as source data, while the destination nodes are elected on the basis of the correlation with the source nodes in the source chain. Additionally, the source and destination nodes can operate in active or passive mode based on their role in their tasks on the network.

  Overall, there seems to be some evidence to indicate that blockchain networks can interoperate and exchange transactions between each other, but will be greatly improved if integrated with IoT systems to allow separation of concerns and business logic between different types of transactions.

- *Variation:* **Blockchain and non-blockchain networks.** This architectural style assumes that resource-rich devices work as connectors to send data transactions from IoT devices to specialized blockchains. Specifically, the connectors create a non-blockchain network that processes transactions before pushing the generated hash to the blockchain via a smart contract. However, resource-rich devices can be grouped on different networks to hold specialized blockchain. It could minimize latency and bandwidth consumption and lead to faster blockchain-related tasks (i.e., mining, transaction processing) [118].

4.5 Evaluation

4.5.1 Qualitative evaluation

We use some ATAM steps to understand the potential tradeoffs and constraints of the candidate styles and evaluate how well they address the quality attribute requirements. The ATAM provides a systematic framework to assess software architectures in terms of design decisions, particularly those that influence the control of quality attribute requirements [65]. Specifically, we use ATAM to refine the quality attributes of our healthcare example into more specific scenarios. We then design our utility tree to prioritize among the qualities of the system and evaluate the suitability of the candidate styles in terms of latency, network usage, energy consumption, and cost.

Quality attribute utility tree

Figure 4.5 shows the utility tree elicited from the most relevant quality attribute requirements of the healthcare example and refined into more specific scenarios. Specifically, the utility tree translates desirable quality attributes into testable scenarios annotated with stimuli and responses, and prioritizes between them [65]. In this analysis, some quality attribute requirements, such as interoperability and availability, have not been considered because of the difficulty of judging them from the context of architecture simulation. In addition, blockchain design decisions that include the consensus mechanism, type of blockchain, and on-chain/off-chain storage have also been mentioned, but they do not have a direct implication in the evaluation.

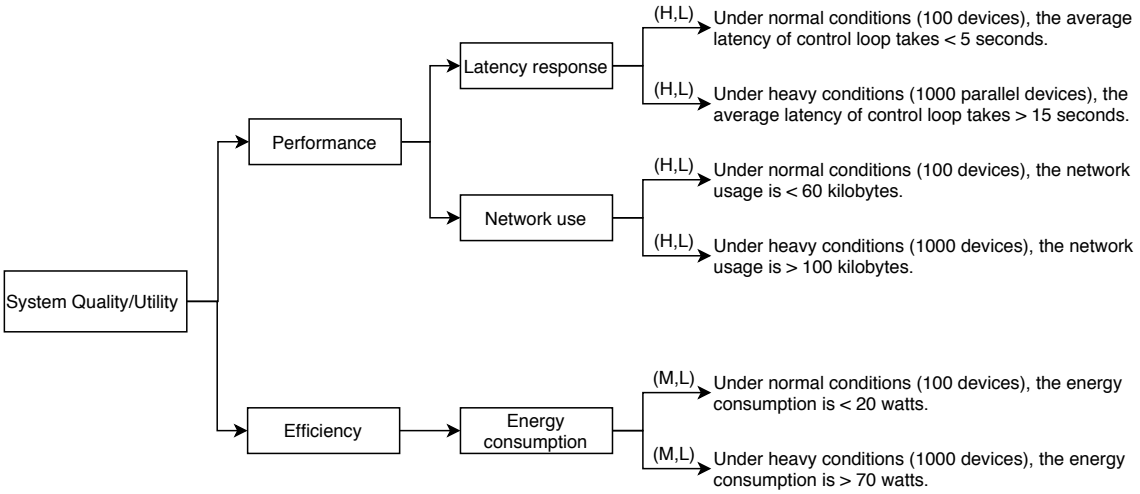

Figure 4.5: Utility tree obtained from the motivation example.

Analysis of the reference architecture styles

We instantiate each of the candidate styles to identify risks, sensitivity, and trade-offs between the quality attribute requirements for the integration of blockchain and IoT. In particular, we associate the highest priority quality attribute requirements with the candidate styles to perform the attribute-specific analysis, as described in Table 4.1. At this point, our analysis has identified three risks, one non-risk, four sensitivities, and three tradeoff points. To better understand the impact of design decisions on each of the candidate styles, we individually analyze them with respect to the selected quality attributes associated with the integration of blockchain and IoT, as shown in Figure 4.6. A single arrow represents the satisfied relationship between a quality attribute requirement and the architectural style. Since a software architecture is the realization of quality attribute requirements that could lead to benefits and tradeoffs [12], we use a plus sign (+) and a minus sign (-) to represent the benefits and tradeoffs of each architectural style, respectively. Such benefits and tradeoffs are followed by the system quality attribute that positively or negatively affects them.

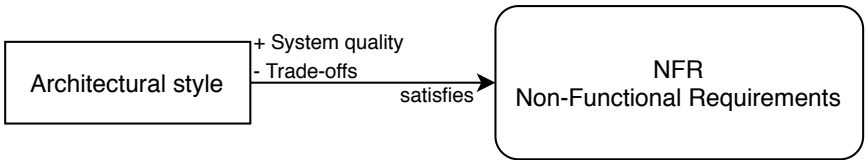

Figure 4.6: Style analysis based on ATAM.

Table 4.1: Risk, sensitivity, and tradeoff points.

Architectural style	**Risk**	**Analysis**
Directly connected IoT-Blockchain	Risk	Because the architecture can allow IoT devices with large computational capacity and battery life to perform blockchain-related tasks, resource-constrained devices (i.e., sensors) cannot connect to the blockchain network.
	Sensitivity	The resource-rich IoT devices can be exhausted with computation (i.e., mining) and data storage loads (i.e., keep a copy of the entire ledger) due to the large deployments of IoT systems.

Continued on next page

Table 4.1 – *Continued from previous page*

Architectural style	**Risk**	**Analysis**
	Tradeoff	* Deploying blockchain in resource-rich devices can lead to fully distributed IoT systems, but the resources in IoT devices can be overloaded. * Processing data in resource-constrained devices the central controller can improve bandwidth efficiency, but availability can be compromised.
Indirectly connected IoT-Blockchain	Risk	Because the architecture can move computation and storage loads from IoT devices to a central controller at the edge, deploying a resource-rich device to handle the growing demands of IoT devices can be costly and lead to security and privacy concerns.

Continued on next page

Table 4.1 – *Continued from previous page*

Architectural style	**Risk**	**Analysis**
	Non-risk	Because the controller instances can be overloaded with data processing tasks, replacing the controller with another instance can require limited effort.
	Sensitivity	A central controller for transaction processing tasks can become a bottleneck as the amount of data increases in the network over time.
	Tradeoff	Processing data in the central controller can improve bandwidth efficiency, but can compromise availability.
Hybrid connected IoT-blockchain	Risk	Because multiple blockchains are deployed for enabling separation of concerns among different type of transactions, the management of the chains and the definition of their security level can become challenging.

Continued on next page

Table 4.1 – *Continued from previous page*

Architectural style	**Risk**	**Analysis**
	Sensitivity	* The deployment of multiple blockchains enhances security in IoT data transactions without affecting the security of other chains. * The number of deployed blockchains might be sensitive to the IoT system's requirements and level of security of the chains.
	Tradeoffs	Enabling multiple blockchains leads to improved security, but comes at the cost of high-energy consumption and computing resources.

- *Analysis of the directly connected IoT-Blockchain style:* Figure 4.7 shows the analysis for selecting architectural style I. This style relies on IoT devices with high computational capabilities to process data before pushing the results to the blockchain without the need of a trusted third-party. This design can lead to improved security because IoT data is processed on the devices themselves and to reduced latency because the data do not need to be sent to the cloud for processing and analysis. However, it assumes that IoT devices have enough computation and connection lifetime to handle complex computation, which is not feasible in reality due to the hardware heterogene-

ity in IoT devices and dynamic traffic in IoT networks [147]. As the amount of data increases over time, IoT devices could become a bottleneck in the network and lead to reduced availability, because data processing tasks will be executed only if IoT devices are available and have enough computational and storage resources. It can also result in reduced resource efficiency since even though the execution of computationally expensive tasks in IoT devices reduces computation and storage loads in blockchain nodes, IoT devices could become overloaded as the amount of IoT data increases over time. As a result, software developers can use this architecture as a starting point to combine blockchain and IoT and deploy applications, such as financial transactions and rental services [143]. For these applications, the integrity of IoT data is mainly dependent on the security of the IoT devices, and transactions take place with low latency.

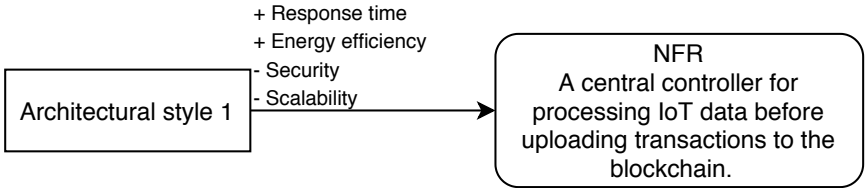

Figure 4.7: Analysis of the Centralized IoT-Blockchain style using ATAM.

- *Analysis of the indirectly connected IoT-Blockchain style:* Figure 4.7 presents the architectural analysis of the indirectly connected style that uses a controller to process IoT data and send it as transactions to the blockchain. As a single controller is used in the end-to-end infrastructure, the implementation cost and power consumption are low. However, due to the large amount of data generated by IoT devices, the controller could be burdened with computation and data storage requirements and lead to a single point-of-failure in the network. Additionally, security could also become a major concern since the controller can be compromised with malicious data uploaded from fake IoT devices. This architecture can be used in smart home [30, 31] and smart

living applications [45] where devices with limited resources sense the environment and transmit sensor readings to a controller that processes and analyzes the data before pushing the results to the blockchain.

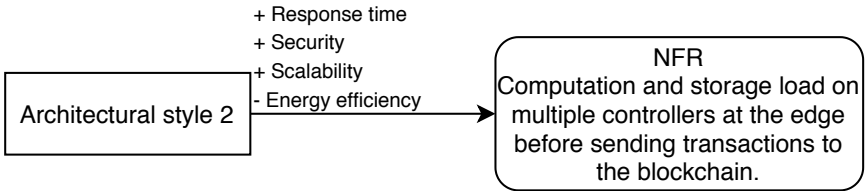

Figure 4.8: Analysis of the Partially decentralized IoT-Blockchain style using ATAM.

- *Analysis of the hybrid connected IoT-Blockchain style:* Figure 4.9 presents the analysis for selecting the architectural style III. This style implements multiple controllers and specialized blockchains to handle IoT data and enable separation of concerns among different types of transactions. Therefore, the implementation of this style could lead to improved security, since if a node is compromised in one of the blockchain networks, it does not affect the other chain. It could also reduce bandwidth and network latency since multiple controllers are deployed at the edge to reduce computation and data storage in IoT devices. However, the implementation cost could be high since multiple controllers are required to meet the computation and data storage demand of IoT systems. Similarly, the deployment of a specialized blockchain could require a large amount of computation and energy power to process transactions and mining blocks.

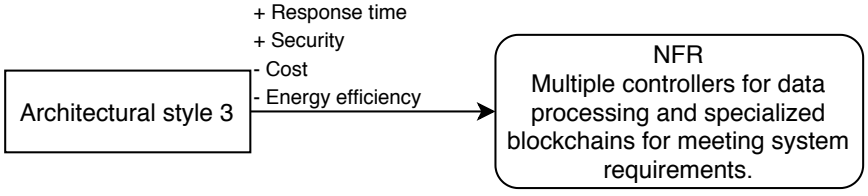

Figure 4.9: Analysis of the Fully decentralized IoT-Blockchain style using ATAM.

4.5.2 Quantitative evaluation

We first simulate the candidate architecture styles for the healthcare example to get quantitative results that complement the qualitative ATAM analysis. Next, we experimentally evaluated the styles to demonstrate their effectiveness in terms of latency, network usage, and energy consumption.

Simulation environment

The evaluation of candidate styles consists of two stages: (1) collect, process and store IoT data in a single or multiple gateways for processing and analysis, and (2) send the IoT data to the blockchain network for immutable storage.

We use Ganache as a personal Ethereum blockchain and Proof-of-Authority (PoA) [89] as the consensus protocol in the simulation of styles. It is important to mention that the IoT devices, gateway(s), and the blockchain nodes are virtualized on an Intel(R) Core(TM) i7-8700 CPU @3.20 GHz 16 GB DDR3 RAM and have the following configurations:

- *IoT device:* Raspbian Buster, 5% of an Intel (R) Core(TM) i7-8700 CPU @3.20 GHz 128 MB RAM; Oximeter data integrated sensor library; Python 3.
- *Controller device:* Linux Mint 19.12, 20% of one Intel(R) Core(TM) i7-8700 CPU @3.20 GHz 512 MB RAM; Python 3.
- *Blockchain:* Linux Mint 19.12, 50% of an Intel (R) Core (TM) i7-8700 CPU @ 3.20 GHz 8 GB RAM, Ethereum 1.7.2 2, and Ganache (test network) 2.1.2 3; Python 3.

We use *nmon* to measure network usage and *power meter* to monitor host machine energy consumption as described in [117].

Experiment setup

We simulated the styles for 1 hour to demonstrate their behavior under different configurations: *Config 1, Config 2, Config 3, and Config 4* having 100, 250, 500, and 1000 transactions. Initially, IoT devices collect oximeter data as 1 MB files and transmit them to a gateway device or multiple gateways, depending on the style. Once data transmission to the gateway is complete, it calculates the hashes and sends them to the blockchain network through a web3 provider, which offers a JSON-RPC API to read and write data to the blockchain [89].

We define the following concrete metrics to evaluate the efficiency of the candidate styles for IoT systems supported by blockchain as follows:

- *Average latency* is measured as the sum of the network propagation delay and the application execution time on each of the interplaying components in the candidate styles.
- *Network usage* is defined as the use of networking resources in each of the intervening components in the candidate styles.
- *Energy consumption* is calculated as the amount of energy consumed by the host machine where the experiments are running.

Result analysis

- *Average latency:* Figure 4.10 shows the average latency of the candidate styles under different configurations. In the case of style II, the controller turned into a bottleneck in the execution of 1000 transactions, which caused a significant increase in latency. In contrast, style I maintains low latency, as IoT devices are directly connected to the blockchain. In the case of style III, the deployment of multiple controllers at the edge

reduces computation and data storage loads on the blockchain.

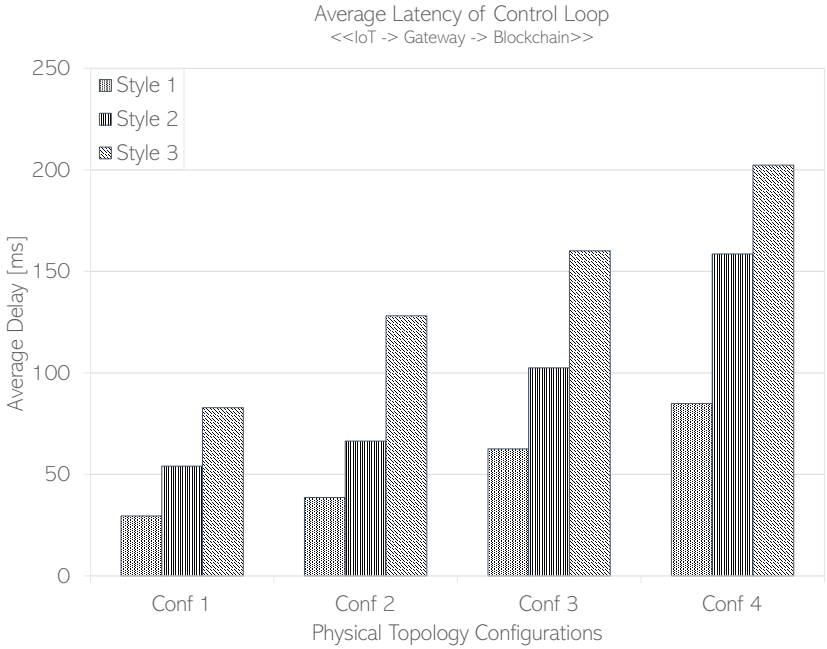

Figure 4.10: Comparison of the average delay of the control loop.

- *Network usage:* Figure 4.11 shows the use of candidate styles in the network in different configurations. In the case of style III, when two or more gateways or blockchain networks are deployed, the load on the network increases significantly. This observation can be attributed to the fact that most of the data communication is performed between IoT devices and the controllers through low-latency links, and hashes of IoT data are sent as transactions to the blockchain.

- *Energy consumption:* Figure 4.12 illustrates the energy consumed for the candidate styles under different configurations. In the case of styles II and III, the gateway(s) at the edge of the network process and analyze IoT data, which drain out a large amount of power. In addition, the implementation of two or more specialized blockchains leads to increased energy consumption in the proposed configurations.

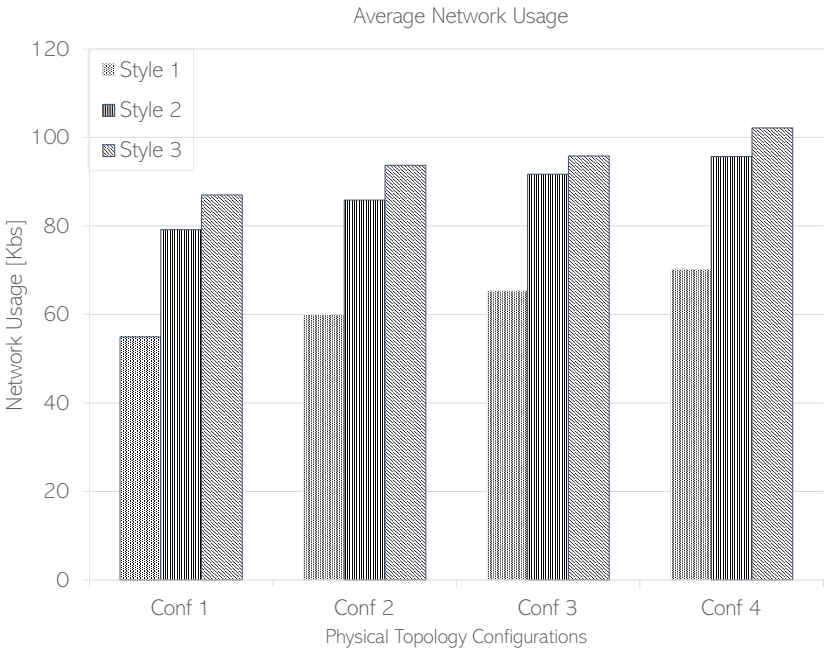

Figure 4.11: Comparison of network usage.

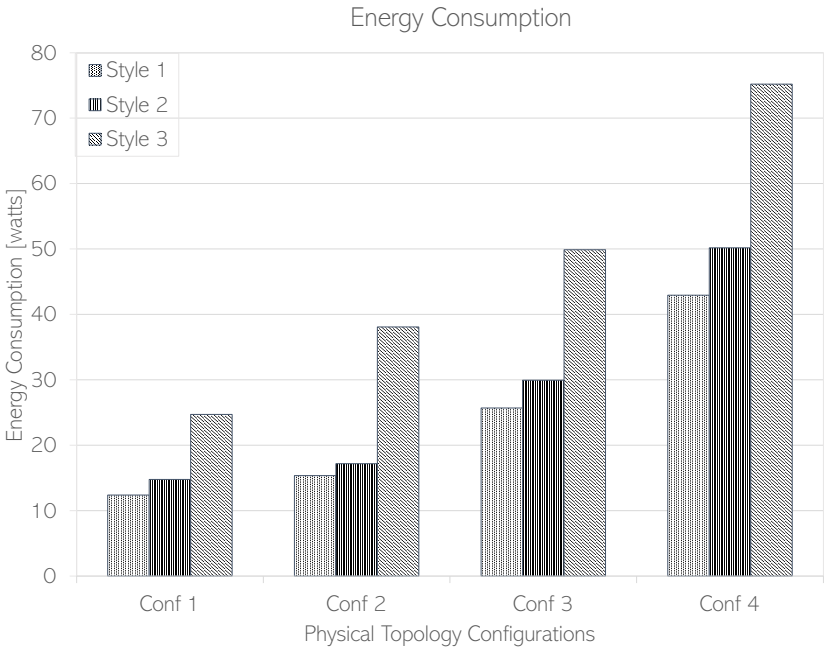

Figure 4.12: Comparison of energy consumption.

4.6 Discussion and Threats to Validity

4.6.1 Discussion

The ATAM evaluation of the reference architecture styles reveals some potential issues in the documentation of the architecture, the collection of the precise quality requirements to be satisfied, and potential architecture tradeoffs. It is worth nothing that we partially use ATAM to understand and document the design tradeoffs in the reference architecture styles. We highlight the most significant problems as follows:

- *Requirements.* The ATAM analysis is used to understand the qualities of the system and to identify new requirements as a result of the evaluation. For example, in the healthcare system, the performance timing requirements correspond to the sum of the propagation delay and application execution time limited to less than 250 ms when 1000 transactions are sent to the blockchain. There is no other performance requirement in the system, such as the time for the deployment of a new fog node or cloud server. These questions are part of the architecture deficiency that was identified during the architecture evaluation process. Furthermore, there was no explicit scalability requirement in terms of the number of IoT devices, fog nodes, and cloud servers required to achieve business objectives. This requirement was identified during the architectural evaluation exercise and should be considered to minimize design risks and lead to improvements in software architecture.

- *Architecture problems.* In addition to the aforementioned risks, sensitivity, and tradeoffs points, we discover some architectural weakness concerning the communication patterns within the candidate architecture styles. In the architecture styles II, the communication pattern could be easily identified by an attacker who discerns data flow and communication between the layers in the architecture. Thus, we assume that

the probability of failure of the central controller increases during the implementation of these styles. Its flaws could be easily mitigated by deploying multiple data controllers that manage data in a cost-effective and efficient manner, which in turn contributes to improving the performance of the system.

4.6.2 Critique of the architecture options

According to the ATAM evaluation, the identified reference styles characterized the quality attribute responses as follows:

- *Directly connected IoT-Blockchain style* shows a high network latency due to computation and storage loads in IoT devices. In this style, IoT devices have to process a large amount of sensor data and calculate the hash of the data to push them to the blockchain network. Although this style is an expensive option for IoT devices, it can be considered as a starting point to enable the integration of IoT and blockchain. Overall, this style fully meets R1 regarding data immutability and integrity and partially addresses R2 in terms of data processing close to the source to minimize latency and bandwidth consumption, but it fails to address R3 related to the efficient utilization of IoT hardware resources.

- *Indirectly connected IoT-blockchain style* exhibits a poor performance and availability problems because it relies on a single controller that could become a single point of failure and bottleneck in the network with the large deployment of IoT systems. It is also the least expensive option in terms of hardware costs, because a single server is required to process IoT service requests and decide on data processing and storage tasks in the blockchain. In summary, this style addresses R1, R2, and R3 in terms of ensuring data integrity, reducing latency, and efficient usage of hardware and software

resources.

- *Hybrid connected IoT-Blockchain style* offers better security, slightly improved availability than style I. However, this solution is computationally expensive in terms of energy consumption and verification and validation requirements compared to style I and style II. By enabling multiple blockchain where tasks are distributed accordingly to the application requirements, i.e., a blockchain for high-speed processing, another for large computation, it affects positively the efficiency and performance of the system. Overall, this style meets R1 in terms of data integrity, but fails to address R2 and R3 with respect to optimal network usage and software/hardware resources.

Based on the ATAM evaluation result, our aim is to create a generic reference approach that combines the aforementioned classical architecture designs and generates the expected system quality goals. The generic style should combine multiple gateways and specialized blockchain to enable separation of concerns among different types of transaction and performing data processing tasks on specific edge networks.

4.6.3 Threats to validity

We carried out the evaluation of our candidate styles through a set of experiments and simulations that resemble IoT systems supported by blockchain. We built our simulation on the real capabilities of IoT devices (i.e., Arduino Yun) and gateways (i.e., Raspberry Pi) according to the setup described in [89]. Although a potential threat to validity is the execution of the experiments in a controller environment, it facilitates faster experimentation in different scenarios that would be expensive to analyze in a real IoT environment. For simplicity, we measure latency, memory usage, and energy consumption. However, our approach can be extensible to multiple quality attributes (e.g., availability, reliability, security, etc.). In ad-

dition, style III has considered the implementation of two blockchains (i.e., local and hosted blockchain, respectively) that operate independently. However, their interaction can have several degrees of complexity and overhead that could require an extensive experimental study.

4.7 Related Work

Several studies propose architectures for the integration of blockchain and IoT, but are not systematic approaches that address the impact of design decisions on the quality attributes associated with IoT systems supported by blockchain. In [30], [31], a lightweight blockchain architecture is proposed to reduce the computation and bandwidth requirements of the classical blockchain. The architecture consists of three layers: smart home layer, overlay ledger (public blockchain), and the cloud, where resource-constrained IoT devices in the smart home are managed centrally by constituting nodes acting as gateway devices. Similarly, in [115], virtual resources are introduced at the edge of the network to deploy IoT applications supported by blockchain. These virtual resources that work as intermediate components facilitate shifting computation and storage loads are offloaded to powerful devices located close to where data is collected. In [132], a blockchain-based architecture is proposed for industrial IoT, where an integration component is implemented to manage the interaction between machines and the blockchain. Moreover, the author uses smart contracts to record transactions and information related to interactions in the industrial environment.

Only a few studies focus on the design issues when architecting IoT systems supported by blockchain from the software architecture perspective. In [79], [78], three reference architecture styles are proposed to facilitate the integration of blockchain and IoT called Fully Decentralized, Pseudo-Distributed Things, Distributed Things, and Fully Distributed. Sim-

ilarly, in [111], three architectures are provided, called IoT-IoT, IoT-blockchain, and hybrid approach, to provide different alternatives as to where IoT interaction can take place. In [114], the use of fog and clouds is analyzed as hosting environments where fog minimizes network latency. Our work can be considered as complementary to the above studies, all of which present different design decisions and architectural approaches to support the integration of IoT and blockchain. Our study differs from [79], [111], [78], as follows: (i) we identify three reference architecture styles and propose variant models for the integration of blockchain in IoT systems, (ii) we use ATAM analysis as a qualitative evaluation method to assess the general fitness of the styles in terms of the identified quality attribute requirements, and (iii) we also perform a quantitative evaluation through simulation to demonstrate the applicability of our approach.

4.8 Conclusion

This work delivers three reference architecture styles for the design of IoT systems supported by blockchain: (i) Directly connected IoT-Blockchain, (ii) Indirectly connected IoT-Blockchain, and (iii) Hybrid connected IoT-Blockchain. To obtain a better judgment of the suitability of the candidate styles, we performed a qualitative and quantitative evaluation following some ATAM and simulation steps, respectively, through an industrial case study. The quality analysis allows us to understand the tradeoff points that need to be considered for given scenarios of interest, while quantitative analysis provides us with concrete results about the identified tradeoff points. The simulation result shows that style II outperforms style I, since the computation and data storage load are performed on a gateway device located between the IoT devices and the blockchain. However, style III shows an increase in performance compared to style II, since the data processing load is split between local gateways. Similarly, style II outperforms style III, since the implementation of two blockchains

requires a large amount of processing and energy resources.

In addition, we identify a gap and opportunity for research in the remainder of this thesis regarding the development of an intelligent mechanism to switch between a blockchain and non-blockchain architecture using self-adaptivity and artificial intelligence algorithms. The mechanism should be able to deal with the inherent limitations of the IoT and blockchain to satisfy the desired quality attributes of the system.

Chapter Five

Data Allocation Mechanism for data-centric IoT systems supported by Blockchain

In chapter 4, we present a set of architecture styles and variants that can be used by software architects and developers as candidate architectures for building IoT systems supported by the blockchain that fulfill the desired qualities. In this chapter, we present a novel data allocation mechanism to decide on on-chain and off-chain storage, considering context information, quality attributes, IoT constraints, and inherent limitations of the blockchain (thereby addressing RQ3). It relies on a controller based on fuzzy logic and context information to decide on which data need to be recorded on the blockchain (i.e., on-chain) or in external storage (i.e., off-chain). To demonstrate the applicability and efficiency of the mechanism, we instance it in two of the architecture styles and variants described in Chapter 4. The results suggest improvements in network usage, latency, and blockchain storage, as well as a reduction in energy consumption.

5.1 Overview

Blockchain enables a decentralized architecture in which IoT data can be recorded as immutable transactions and processed by consensus from the blockchain nodes [147]. Due to the inherent features of blockchain, such as transparency, auditability, traceability, and accountability, it can improve the management of distributed and secure IoT data [89]. However, data management cannot be fully achieved in practice by simply combining IoT and blockchain. IoT systems are data-driven in nature, characterized by high velocity, high volume of data, and high mobility, making data security and its management a challenge. On the other hand, blockchain presents technical constraints of a complex nature that can limit its adoption in IoT systems at scale, such as limited space, immutability, and excessive computational power, among others. Therefore, developing a mechanism for dynamic data management and its allocation that deals with the constraints of IoT and the technical limitations of blockchain would be of great value for designing IoT systems supported by blockchain.

Many research approaches have reviewed the application of blockchain in IoT systems. The goal of these approaches is to rely on the inherent features of blockchain to overcome the challenges of IoT. For example, in [133] a food traceability supply chain system is proposed using RFID and blockchain to improve food safety and reduce logistic costs. In another work [75], an energy trading system is developed using consortium blockchains to reduce trading costs by allowing distributed consensus. Similarly, in [35] a blockchain-based healthcare platform is presented to improve privacy in healthcare data management. In [75], a blockchain-based healthcare care platform is developed to facilitate clinical trials and precision medicine. However, these approaches suffer from two drawbacks: first, they are designed in an ad-hoc manner; thus, they can be used for developing a particular IoT application. Second, they do not deal with the IoT constraints and technical limitations of the blockchain, which can

affect the management of IoT data and its allocation in the blockchain. Therefore, there is a need for dynamic data allocation mechanisms for IoT systems supported by blockchain to decide on on-chain and off-chain data allocations, considering context information, quality attributes, IoT constraints, and blockchain limitations.

This chapter develops a novel data allocation mechanism to dynamically decide on on-chain and off-chain data storage, considering context information, quality attributes, IoT constraints, and blockchain inherent limitations. It implements a controller based on fuzzy logic and context information to decide on which data needs to be recorded in the blockchain (i.e., on-chain) or in external storage (i.e., off-chain). In particular, the blockchain is used to record small critical data that need to be securely shared among participants in the system, while the external storage (e.g., cloud, fog, or local database) keeps the raw and non-critical data. The mechanism operates as follows. First, the controller extracts context information from IoT data (e.g., data, network, and quality). Here, data refers to raw data collected by IoT devices, network corresponds to the number of points of exchange interested in IoT data, and quality refers to accurate measurement of the device itself. Next, the controller translates crisp context information (i.e., bits) to fuzzy inputs (i.e., severe, moderate, etc.) using the membership functions defined by the domain expertise to get the fuzzy output. Finally, a set of membership functions is defined to map the fuzzy output to a machine-readable value, which is used as a threshold value called Rating of Allocation (RoA) to make allocation decisions.

To demonstrate the flexibility of our approach, we instantiate the mechanism in two commonly used architecture styles for IoT systems supported by blockchain, i.e., blockchain-based cloud and fog [114, 77]. Using blockchain in fog and cloud, it provides additional security to the two computing environments by ensuring the immutability, traceability, and integrity of the data [137]. However, the design and realization of the data allocation mechanism lead to refinements of the existing architecture styles, which should consider the QoS requirements of IoT systems and the constraints imposed by the hosting environments, e.g.,

fog and cloud. To this end, we envision a four-tier abstraction, i.e., the IoT tier, the data controller tier, the fog tier, and the cloud tier, where the data controller tier is introduced between the IoT tier and the fog tier. The data controller tier enables the data allocation mechanism to decide which data needs to be stored within the blockchain embedded in the fog or the cloud or allocated off-chain (e.g., cloud database).

We compare our approach with some existing decision-making mechanisms, such as logistic regression [83, 156] and the decision tree [97, 82] to evaluate how they perform in terms of CPU usage and execution time. We also evaluate the refined blockchain-based cloud and fog architectures in the above healthcare case study using the FogBus framework [137]. First, the mechanism runs on the refined blockchain-based cloud and then is executed on the refined blockchain-based fog. The evaluation focuses on enabling and disabling the data allocation mechanism in the two IoT-blockchain architecture styles and compares their performance in terms of blockchain size, latency, energy consumption, and network usage. The results show that latency is reduced by 36% in the refined blockchain-based cloud and by approximately 27% in the refined blockchain-based fog. Similarly, energy consumption is reduced by an average of 28% in the refined blockchain-based cloud and fog. Furthermore, network usage is reduced by 32% in the refined blockchain-based fog and 24% in the refined blockchain-based cloud. This chapter presents the following.

- A novel data allocation mechanism to decide which data needs to be recorded in the blockchain or external storage using context information and a fuzzy logic mechanism.
- A refinement of two architecture styles for data-centric IoT systems supported by blockchain, i.e., blockchain-based cloud and fog, as a result of the implementation of the data allocation mechanism.
- An evaluation of the refined blockchain-based cloud and fog by applying them to a health-care case study using FogBus. The experimental results suggest significant

improvements in data transaction latency, network usage, energy consumption, and blockchain storage usage.

The remainder of this chapter is organized as follows. Section 5.2 presents a motivation scenario and its architectural requirements for the development of IoT-blockchain systems. Section 5.3 models the proposed data allocation mechanism. Section 5.4 introduces the refined blockchain-based cloud and fog architectures. Section 5.5 provides an illustrative example that supports the implementation of the data allocation mechanism in the two commonly used IoT-blockchain architecture styles. Section 5.6 conducts a set of experiments to evaluate the effectiveness of the data allocation mechanism and compares our approach with other existing decision-making mechanisms. Section 5.7 summarizes the related work in fuzzy logic and IoT. Section 5.8 presents the envisioned challenges and possible future research, and Section 5.9 concludes the chapter.

5.2 Motivation Example and Requirements

This chapter uses the same healthcare case study described in Section 4.2.1. For this chapter, we have considered that due to the limited capabilities of the oximeter, it is connected to Alice's smartphone to send the collected data to the fog and cloud infrastructures. We assume architects implement a blockchain in both environments (e.g., fog and/or cloud) to protect Andrew's health data and secure share it with health-care providers (i.e., doctors, hospitals, pharmacies, laboratories, health insurance companies, etc.). If Andrew's is moved from one hospital to another, the uncertainty in the normal range for test results could make it difficult for medical staff to diagnose the disease. Additionally, incomplete and missing information on his health history makes decisions more complex and uncertain.

5.2.1 Architecture Significant Requirements

In addition to the quality requirements of the healthcare case study described in Section 4.2.1, we present the data requirements. We argue that IoT systems can be subject to a variety of uncertainties in their operating environment, such as changes in the traffic network and interference [141, 54]. These uncertainties could lead to incomplete, imprecise, and missing information that makes it difficult to provide accurate decision support [46, 86]. In contrast, many real-world problems require essentially multifactor consideration at the same time before making decisions [26]. In particular, there may be a number of real-world scenarios that cannot be simply analyzed/depicted by a set of binary values if they depend on various factors to make decisions [91, 99, 39]. For example, in a cold supply chain system, instead of simple binary descriptions "cold or hot", indoor and outdoor conditions, such as various levels of temperature / humidity, must be considered to optimally maintain frozen food products under the complex threshold policies [89]. Similarly, in our example of sleep apnea, oximeter data (that is, heart rate and oxygen saturation in the patient's blood) could not be sufficient to understand its criticality or sensitivity levels to maintain its management and allocation soundness. Furthermore, we rely on the blockchain to improve the security and privacy of health data and securely share them between interested healthcare providers [151, 150]. We summarize the requirements that support the adoption of fuzzy logic and blockchain through the health-care example in our approach as follows.

- Requirement 1 (R1): *The approach shall cope with uncertainty and imprecise information.* Patients and / or their families often do not express their symptoms accurately and instead use ambiguous terms that could lead to many suboptimal and even incorrect medical decisions.

- Requirement 2 (R2): *The approach shall cope with missing and incomplete information from sensors,* caused by heterogeneous hardware and software in IoT devices and

dynamic traffic in IoT networks (i.e., devices join and leave the network).

- Requirement 3 (R3): *The approach shall consider the cases with more than one decision-making dimensions.* In particular, multiple sensor readings and environmental information must be collected to diagnose patient conditions and provide accurate treatment.

- Requirement 4 (R4): *The approach shall rely on decentralized infrastructures for secure data storage and data sharing among interested parties.* In particular, IoT systems tend to shift from centralized infrastructures to record data in a decentralized fashion and empower users with control over their records.

The application of fuzzy logic and blockchain fits the above requirements on the uncertainty of data values *R1, R2, R3* and the security requirements of critical IoT applications *R4*. To meet these requirements, we design a data controller based on fuzzy logic that extracts multiple context parameters of each data request, i.e., data, network, and quality, to calculate the Rating of Allocation (RoA). This value gives us insight into the sensitivity of IoT data to decide which data request needs to be stored in the blockchain or allocated off-chain (e.g., cloud database). In the context of our study, the context parameters, i.e., data, network, and quality, are used as inputs to the proposed mechanism where *data context* refers to sleep apnea levels, i.e., mild, moderate, and severe [137] which could be used by malicious parties to infer users profile. *Network context* relates to the number of sharing points in the health-care network interested in the collected data, e.g., doctors, hospitals, pharmacies, laboratories, health-care insurances, etc. [32]. *Quality context* corresponds to precision measurements of the device that must be protected to ensure reliable medical analysis [4]. We compute the three parameters to derive the RoA value and determine whether a particular data request needs to be allocated within the blockchain or kept off-chain. Additionally, we enrich two existing IoT-blockchain architecture styles (i.e., blockchain-based cloud and fog) with data control and data management capabilities to support on-chain data

allocation. Specifically, we introduce a data controller tier between the IoT and fog tiers to handle on-chain or off-chain data allocation decisions based on the RoA value.

5.3 Data Allocation Mechanism based on Fuzzy Logic

In this section, we provide a brief introduction to Fuzzy Logic and motivate our approach. Next, we explain the proposed data allocation mechanism and the calculation of the RoA value to support data allocation.

5.3.1 Fuzzy Logic

Fuzzy Logic is an Artificial Intelligence (AI) technique that uses linguistic variables to mimic human thinking and enable decision-making in real-time systems [8]. This approach aims at solving problems that are difficult to formulate mathematically due to imprecise or non-numerical information, such as “very cold" or “not very satisfied" [3]. Unlike the classical one-to-one input-to-output control strategy, fuzzy logic makes decisions out of many-to-one and many-to-many input-to-output control [88] by using fuzzy sets and rules [90, 93]. The fuzzy logic process consists of three stages: fuzzification, inference rules, and defuzzification, as shown in Figure 5.1.

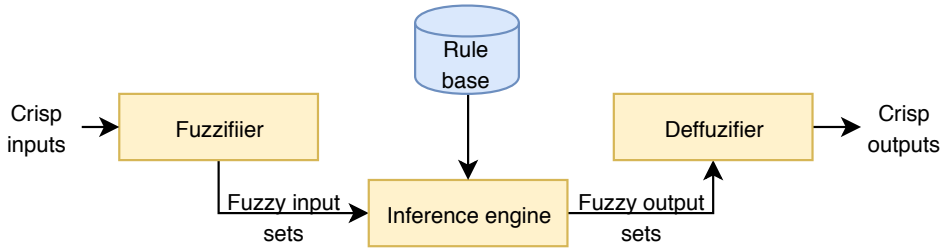

Figure 5.1: Fuzzy logic process [8].

1. *Fuzzification* converts crisp input data collected by the sensors (i.e., bits) to a fuzzy input set of linguistic terms using the membership functions.

2. *Inference* applies a set of IF-THEN rules defined by domain experts to derive fuzzy output.

3. *Defuzzification* maps the fuzzy output to a crisp machine-readable output using the defined membership functions.

5.3.2 Rationale behind the adoption of Fuzzy Logic and Blockchain

IoT networks are subject to changes in operational contexts, such as dynamic traffic and interference [141, 54], which can lead to uncertainties in data values, management, and allocation. These uncertainties mainly caused by the volatility of the network and changes in connectivity can cause several issues, e.g., data inconsistency, incompleteness, imprecision, and / or vagueness [46, 86]. If we apply only coarse-grained representations, that is, true and false, to depict system features / outcomes, we would end up with superficial understandings/decisions [91] of the systems. In particular, binary logic deals with two possible values, 0 (false) and 1 (true). For example, to make an air conditioner decision based on indoor temperature, if the temperature hits above

$$30°C$$

, then turn on the cooler mode. Otherwise, if the temperature hits below

$$18°C$$

, then turn on the heater mode. In general, we can infer that binary logic is suitable for scenarios in which solutions are made binary under policies that exhibit certainties with reliable sensing and affirmative values, so that data management and its outcome can be reliably predicted [91].

Moreover, in dynamic/adaptive systems, e.g., AI-based systems, the allocation policies and decisions could be even more complex to establish before execution and therefore it is infeasible to simply apply “true or false" to represent the states of the system [26]. For example, fuzzy logic can be used in a fire detection system to identify whether there is a fire or not, but also to analyze the intensity of the flames [80]. The state of the system can also change in different contexts; for example, what is true in one context may be false in another [86]. For example, fuzzy logic can be used in a smart home scenario to start the AC and fan according to the environment and the room temperature, where the fan only works if there is movement in the room [99]. Despite the use of fuzzy logic in real scenarios, it has two major limitations: (1) it depends on human knowledge and expertise, and (2) the efficiency of the system is not high because it handles imprecise and inaccurate data as input. To address these issues, many studies have considered the use of fuzzy neural networks (FNN) to make decisions in IoT systems and overcome the need for prior knowledge to create inference rules [16]. FNN combines fuzzy logic and neural networks to perceive patterns from input, which are used to create the rules of the fuzzy system. For example, FNN is used in an IoT-based healthcare system to determine serious diseases and make decisions accordingly [47]. Similarly, blockchain and adaptive neuro-fuzzy inference system (FS-ANFIS) are used to develop a secure healthcare application for diabetic and cardio disease where a rule-based clustering algorithm is implemented on patient health records and feature selection based on FS-ANFIS to predict the diseases. In general, these studies highlight the need for a priori neural networks to make decisions in blockchain and IoT systems [125].

In this work, we use fuzzy logic to make decisions in IoT systems supported by blockchain in the face of incomplete data based on the concept of degrees of truth and true or false. The proposed algorithm makes granular decisions based on multiple states at the same time, rather than dealing with two states [26, 91]. Specifically, we propose a context-aware mechanism based on fuzzy logic that considers context information to optimize

on-chain allocation decisions given changes in operating context and internal dynamics in IoT systems supported with blockchain. Our fuzzy strategy considers three context parameters as input, called data, network, and quality, to decide on which data need to be recorded on-chain or kept in an external storage (i.e., cloud database). Each parameter consists of three membership functions, and fuzzy rules are used to get a particular output to the given input for the system. Furthermore, sensor data is likely to include sensitive and critical information that could be manipulated and altered by untrustworthy service providers in the cloud and lead to loss of data and financial damage [95]. When blockchaining data sensed by IoT devices, architects should also consider that the computation and storage space in the blockchains remain limited [150]. For example, public blockchains can handle on average 3-20 transactions per second while VISA [1] can support around 1700 transactions per second. Therefore, it is essential to develop an efficient data allocation mechanism that copes with uncertainty in data management for IoT systems supported with blockchain. We propose a fuzzy logic-based context-aware mechanism that extracts context information from data, network, and quality to make optimal on-chain allocation decisions. Using fuzzy logic, we aim to minimize the risk of uncertainty, vagueness, and interpretation of incomplete data and offer appropriate allocation decisions in IoT systems supported with blockchain.

5.3.3 Envisaged contexts

Context is defined as the computational representation of any information that can characterize the status of an entity, for example, a user, device, software application, or any other object that handles the interaction between users and services [1]. In this study, the context parameters, i.e., data, network, and quality, are modeled as follows.

- *Data context* is represented as α and refers to the device data, e.g., heart beat rate,

[1] https://usa.visa.com/run-your-business/small-business-tools/retail.html

oxygen saturation, etc. [137].

- *Network context* is represented as β and corresponds to the number of the sharing points interested in the collected data [32].
- *Quality context* is represented as ω and refers to the device accuracy measurements [92].

5.3.4 Data controller structure

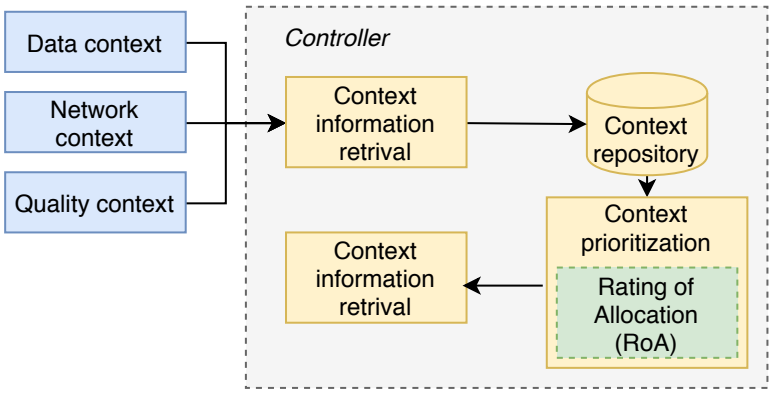

Figure 5.2: Data controller components.

Figure 5.2 shows the components of the data controller, such as context information retrieval, context repository, context prioritization, and context allocation decision.

- *Context information retrieval* extracts the context parameters, i.e., data, network, and quality from the IoT devices and the sharing points interested in the collected data.
- *The context repository* temporarily stores the retrieved context parameters before moving them to the context prioritization component.
- *Context prioritization* computes the RoA value based on the context parameters and sharing points.

- *Context allocation decision* uses the RoA value as a threshold measurement to determine which data request needs to be allocated within the blockchain or stored off-chain.

Figure 5.3 illustrates the proposed data allocation mechanism, and Table 5.1 defines the relevant notation.

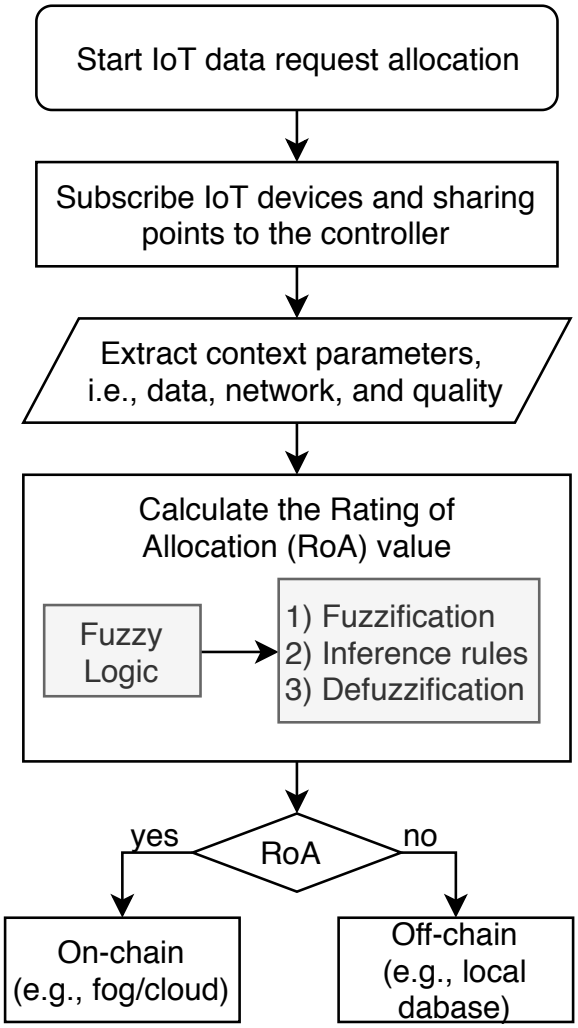

Figure 5.3: Flowchart of the data allocation mechanism.

The data allocation mechanism is initialized by subscribing IoT devices and sharing points interested in the collected data with the data controller. After subscription, *context information retrieval* extracts the context parameters $E_{d_r} \in \left\{U_{\alpha}^{d_r}, U_{\beta}^{d_r}, U_{\omega}^{d_r}\right\}$ of each data

Table 5.1: Notations

Symbol	Definition
R	Set of all data requests d_r.
E_{d_r}	Context parameters within a data request d_r.
α	Data context.
β	Network context.
ω	Quality context.
$U_i^{d_r}$	Context parameters (value) of i_{th} for application $d_r \in R$.
δ_{d_r}	Rating of Allocation (RoA) of a data request d_r.
μ_i	Fuzzy membership function for any context parameters E_{d_r}.
F_c	Fuzzy output set for RoA calculation.
f_{d_r}	Fuzzy output in fuzzy output set F_c.
ϕ^{d_r}	Singleton value for a fuzzy output $f_{d_r} in F_c$.
μ_o	Membership function for any fuzzy output f_{d_r} in RoA calculation.
D_s, S_p, D_q	Fuzzy sets for data sensitivity, sharing points, and data quality.

request d_r from a set of data requests R. The context parameters in E_{d_r} are stored in *context repository* and sent to the *context prioritization* component for processing and analysis. Since each context parameter in E_{d_r} uses different ranges and scales, equation 5.1 is used to ensure that the numerical values of the context parameters are normalized in the range 0 to 1.

$$\overline{U_i^{d_r}} = \frac{U_i^{d_r} - \gamma_i}{\lambda_i - \gamma_i} \tag{5.1}$$

$U_i^{d_r}$ corresponds to the numerical value i_{th} of a data request d_r defined in the range, $[\gamma_i, \lambda_i]$ which is set according to the range of context parameters defined in Table 5.2. For example, the data context (α) derived from the level of sleep apnea is represented in a range of 5 to 40 [137]; the network context (β) related to the number of sharing points interested in the collected data is represented in a range of 1 to 5 [32]; and the quality context (ω) referring to the accuracy of the device is represented in a range of 0.1 to 1 [92]. If numerical values of any context parameters do not fit within the defined ranges, the data request is discarded from placing on the blockchain and allocated to an external storage.

Table 5.2: Scope of context parameters.

Parameters	Values
$[\gamma_\alpha, \lambda_\alpha]$	5 to 40
$[\gamma_\beta, \lambda_\beta]$	1 to 5
$[\gamma_\omega, \lambda_\omega]$	0.1 to 1

Next, a fuzzy logic approach is used to build a data controller that calculates the RoA value of each data request represented as δ_{d_r} based on the normalized context parameters in E_{d_r}. In the fuzzification phase, the normalized value $\overline{U_i^{d_r}}$ of any context parameter in E_{d_r} is converted into a fuzzy input set using the corresponding membership functions μ_i. Here, the membership functions of the collected context parameters, for example, data, network, and quality, are applied to three fuzzy sets, i.e., data sensitivity, sharing points, and data quality, as illustrated in Figure 5.4. Each fuzzy set is defined within a normalized range of 0 to 1 as follows:

- Data sensitivity: $D_s \in \{Mild, Moderate, Severe\}$
- Sharing points: $S_p \in \{Small, Regular, Large\}$
- Data quality: $D_q \in \{Poor, Standard, Rich\}$

The membership degree $\mu_i\left(\overline{U_i^{d_r}}\right)$ for any normalized value $U_i^{d_r}$ in the corresponding fuzzy set can be graphically represented as a triangular waveform, trapezoidal waveform, etc. [8]. Here, the trapezoidal waveform is used to represent the dynamic variation of the context parameters in the IoT system. Each membership function has a grade from 0 to 1 at each end point and uses a label to identify its condition.

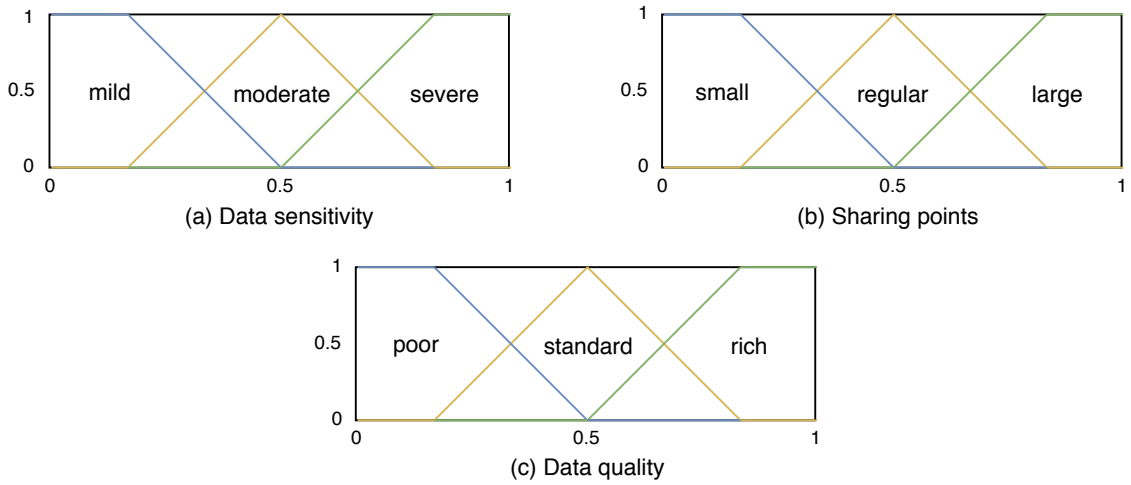

Figure 5.4: Membership functions of the context parameters (a) data sensitivity, (b) sharing points, and (c) data quality.

In the fuzzy inference phase, the data controller evaluates the fuzzy input data according to the IF-THEN rules of the domain expert, where IF captures the system's knowledge using a condition and THEN derives the corresponding fuzzy output as a conclusion. These domain-specific rules allow one to compare the relation between multiples of input and output parameters. Figure 5.5 illustrates a set of fuzzy rules with their corresponding fuzzy output set to calculate the RoA value defined as $F_c \in \{Low, Medium, High\}$.

The following are some representative examples of fuzzy rules to determine the fuzzy output $f_{d_r} \in F_c$ for a data request, d_r used by the data controller to calculate the RoA value.

- **IF** data sensitivity (α) is severe AND sharing points (β) are large AND data quality (ω) is rich, **THEN** RoA is high.

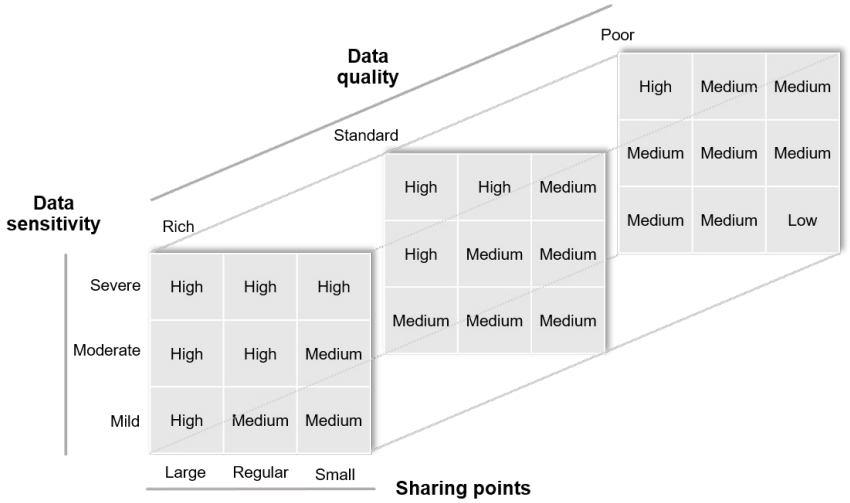

Figure 5.5: Fuzzy rules for the RoA calculation.

- **IF** data sensitivity (α) is normal AND sharing points (β) is regular AND data quality (ω) is rich, **THEN** RoA is medium.
- **IF** data sensitivity (α) is mild AND sharing points (β) is small AND data quality (ω) is poor, **THEN** RoA is low.

In addition to evaluating the rules, the inference phase also combines the results of each rule to determine the fuzzy output. According to the fuzzy rules, severe data sensitivity (e.g., rigid parameter) is given higher weight compared to regular sharing points and standard data quality (e.g., relaxed parameters), since data sensitivity could be used to infer user profile and perform malicious attacks. As a result, the RoA value becomes more aligned with the data sensitivity parameters than the other relaxed parameters. The context parameters in a data request are logically linked through the *AND* operator to deliver the fuzzy output. This operator represents the intersection of membership functions whose values for each context parameter are defined as the *minimum* of individual membership functions [8]. Equation 5.2 is used to calculate the fuzzy output membership function $\mu_o(f_{d_r})$ for a data request d_r.

$$\mu_o\left(f_{d_r}\right) = min\left(\mu_\alpha\left(\overline{E_\alpha^{d_r}}\right), \mu_\beta\left(\overline{E_\beta^{d_r}}\right), \mu_\omega\left(\overline{E_\omega^{d_r}}\right)\right) \tag{5.2}$$

Based on the context input parameters, multiple rules can be triggered at the same time, which requires combining the membership functions of the associate fuzzy output to derive the final result. In the defuzzification phase, the fuzzy output is mapped to a crisp machine-readable output using the defined membership functions. Here, the RoA value δ_{d_r} of a data request d_r is calculated by combining the membership functions of the fuzzy output and using a set of singleton values to distinguish different outputs. For each fuzzy output f_{d_r}, there is a singleton value $\phi_k^{f_{d_r}}$ that is defined as the maximum data request rate for the fuzzy output f_{d_r}. Equation 5.3 calculates the deffuzified RoA value denoted as δ_{d_r} using the Discrete Center of Gravity method as follows:

$$\delta_{d_r} = \frac{\sum_{n=1}^{n=k} \mu_o\left(f_{d_r}^k\right) * \phi_k^{f_{d_r}}}{\sum_{n=1}^{n=k} \mu_o\left(f_{d_r}^k\right)} \tag{5.3}$$

Here, δ_{d_r} corresponds to the RoA value for a data request d_r after applying fuzzy logic on the context parameters in E_{d_r}. Next, δ_{d_r} is used by the *context allocation decision* component to derive the allocation decision for a data request d_r. In particular, when the context parameters of a data request E_{d_r} are higher than δ_{d_r}, then it is allocated within the blockchain; otherwise, it is stored off-chain (e.g., cloud database).

5.4 Data allocation mechanism in IoT-Blockchain architectures

Many studies have focused on the implementation of blockchain in fog and cloud environments to improve its security in terms of immutability, traceability, and data integrity [114,

77]. In [77], a blockchain-based cloud framework is proposed where cloud servers (i.e., application servers, data servers, etc.) become trusted nodes that support IoT data transactions in a distributed and secure manner. Furthermore, in [114] a blockchain-based fog is designed to ensure that the fog nodes are tamper-proof and that the data on them cannot be manipulated or altered by untrusted parties. Despite interest in embedding blockchain in fog or cloud, there is still a need to improve blockchain-based cloud and fog architectures with data management and allocation capabilities to alleviate the storage capacity of blockchain. To this end, we propose a data allocation mechanism that calculates the RoA value of each data request based on multiple context parameters and decides its allocation on-chain or off-chain. The implementation of the mechanism leads to refinements in the IoT-blockchain architecture styles that should reflect the way the mechanism is integrated into them, considering the QoS requirements and the constraints of cloud and fog environments. To handle these refinements, we introduce a data controller tier between the IoT tier and the fog tier that handles data allocation decisions as shown in Figure 5.6, with the details of all tiers illustrated as follows.

- *IoT tier* consists of sensors and actuators that perceive information from the environment and act on the collected data. As many IoT devices have limited computing capabilities to preprocess real-time data, they are connected to proximate gateways to transmit the collected data to the upper levels.

- *Data controller tier* acts as a broker interface between the IoT tier and the fog tier and consists of a network of gateway nodes that implement the data controller logic. The data controller extracts context parameters of each data request, e.g., data, network, and quality, to calculate the RoA value and determine its allocation within the blockchain or off-chain, e.g., cloud database.

- *Fog tier* enables a network of distributed nodes with advanced capabilities (e.g., gate-

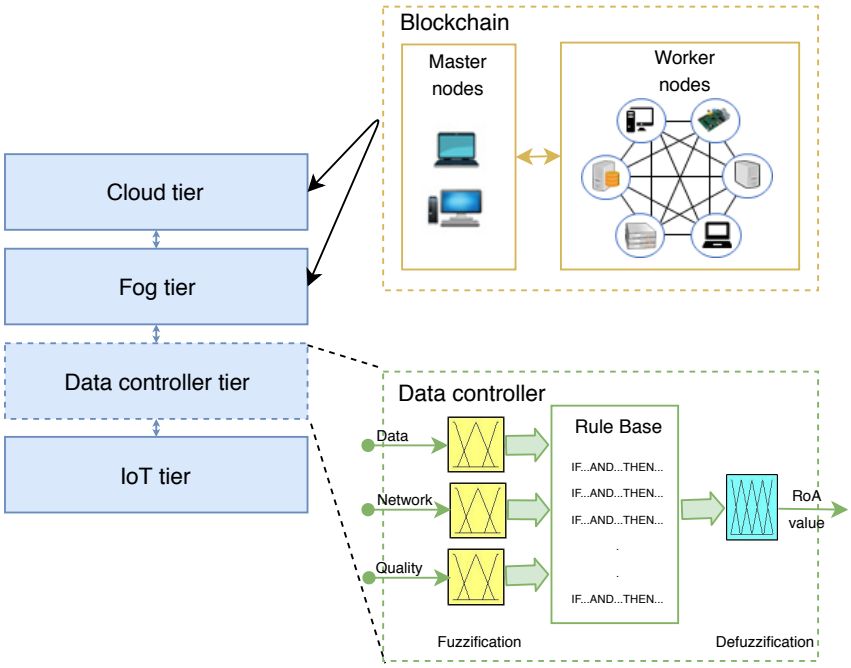

Figure 5.6: Enrichment of the IoT-blockchain architecture styles.

ways, switches, local servers, etc.) that ensure quick processing and short-term storage close to where the data is collected, reducing the amount of data sent to the cloud.

- *Cloud tier* enables a centralized and scalable platform with significant processing and storage resources to support the deployment of IoT applications at minimal cost.

As our approach relies on enriching two commonly used IoT-blockchain architecture styles, i.e., blockchain-based cloud and fog, the blockchain network is designed as follows. *Blockchain* consists of a network of heterogeneous nodes in terms of processing, storage, and energy resources that play different roles in the architecture, e.g., master and worker nodes. This tier can be implemented across diverse computing infrastructures, e.g., fog and cloud, to reduce the overhead in the network and enable secure data sharing in both hosting environments.

1. *Master nodes* receive data requests as transactions from the data tier and discover

worker nodes that can process them in a distributed manner. Moreover, the master nodes generate a public/private key pair to sign the received transactions before broadcasting them on the blockchain network. These nodes also create blocks to store confirmed transactions and calculate the hash of each block to append it to its chain.

2. *Worker nodes* use the public key provided by the master nodes to verify whether the received transaction comes from a legitimate source. Once the transaction is verified, it is validated by consensus following a mining process to be considered confirmed.

5.5 Illustrative example

We explain step by step the proposed data allocation mechanism using the health-care example described in Section 4.2.1. Here, the data controller receives between 20 and 200 data signals per data request from IoT devices and sharing points at any time t. For each data request, the data controller extracts context parameters, i.e., data, network, and quality, where the data context refers to the level of sleep apnea gathered from the pulse oximeter in a range of 5 to 40, the network context corresponds to the number of sharing points interested in the user data (i.e., doctors, hospitals, pharmacies, insurance companies, etc.) in a range of 1 to 5, and the quality context relates to the measurement of the accuracy value provided by the device itself in a range of 0.1 to 1. The data controller then normalizes the context parameters to calculate the RoA value and supports on-chain allocation decisions accordingly. Table 5.3 illustrates representative examples of context parameters that serve as input to the data controller, along with their normalized values and outputs of RoA values. In addition to these parameters, the singleton values are defined as $\phi_{High} = 10, \phi_{Medium} = 5, \phi_{Low} = 2$ to make a clear distinction between intermediate levels of the fuzzy output set F_c defined as *Low, Medium, and High*. These values are associated with the degree of membership of a particular fuzzy set and are defined in the same order as described in Section 5.3.4.

Table 5.3: Parameters of data requests.

Id	α	β	ω	δ
Req1	$E_{\alpha}^{1} = 40$	$E_{\beta}^{1} = 5$	$E_{\omega}^{1} = 1$	8.31
	$\overline{E_{\alpha}^{1}} = 1$	$\overline{E_{\beta}^{1}} = 1$	$\overline{E_{\omega}^{1}} = 1$	
Req2	$E_{\alpha}^{4} = 30$	$E_{\beta}^{4} = 3$	$E_{\omega}^{4} = 0.8$	8.07
	$\overline{E_{\alpha}^{4}} = 0.7436$	$\overline{E_{\beta}^{4}} = 0.5$	$\overline{E_{\omega}^{4}} = 0.7778$	
Req3	$E_{\alpha}^{3} = 35$	$E_{\beta}^{3} = 3$	$E_{\omega}^{3} = 1$	8.27
	$\overline{E_{\alpha}^{3}} = 0.8718$	$\overline{E_{\beta}^{3}} = 0.5$	$\overline{E_{\omega}^{3}} = 1$	
Req4	$E_{\alpha}^{5} = 15$	$E_{\beta}^{5} = 1$	$E_{\omega}^{5} = 1$	5
	$\overline{E_{\alpha}^{5}} = 0.359$	$\overline{E_{\beta}^{5}} = 0$	$\overline{E_{\omega}^{5}} = 1$	
Req5	$E_{\alpha}^{2} = 30$	$E_{\beta}^{2} = 3$	$E_{\omega}^{2} = 0.5$	7.52
	$\overline{E_{\alpha}^{2}} = 0.7436$	$\overline{E_{\beta}^{2}} = 0.5$	$\overline{E_{\omega}^{2}} = 0.4444$	
Req6	$E_{\alpha}^{6} = 38$	$E_{\beta}^{6} = 3$	$E_{\omega}^{6} = 0.6$	8.19
	$\overline{E_{\alpha}^{6}} = 0.9487$	$\overline{E_{\beta}^{6}} = 0.5$	$\overline{E_{\omega}^{6}} = 0.5556$	
Req7	$E_{\alpha}^{7} = 31$	$E_{\beta}^{7} = 4$	$E_{\omega}^{7} = 0.7$	7.91
	$\overline{E_{\alpha}^{7}} = 0.7692$	$\overline{E_{\beta}^{7}} = 0.75$	$\overline{E_{\omega}^{7}} = 0.6667$	
Req8	$E_{\alpha}^{8} = 20$	$E_{\beta}^{8} = 5$	$E_{\omega}^{8} = 1$	7.694
	$\overline{E_{\alpha}^{8}} = 0.4872$	$\overline{E_{\beta}^{8}} = 1$	$\overline{E_{\omega}^{5}} = 0.4444$	

The result in Req1 reveals that RoA is high (8.31) when the sensitivity of the data is 40, the number of sharing points is 5, and the quality of the data is 1. From Req5, we realize that the RoA value is high (7.52) when the data sensitivity is 30, the sharing points is 3 and the data quality is 0.5. However, in Req4 the RoA value is low (5) when the sensitivity of the data is 15, the number of sharing points is 1, and the quality of the data is 1. On the basis of the findings, we conclude that when the data is highly sensitive and its quality is good, the number of the sharing points interested in that particular data increases in the system.

Consequently, we define 7.5 as a threshold measurement to consider data requests with severe data sensitivity, regular sharing points, and standard data quality as the ones to be allocated within the blockchain embedded in cloud and fog environments. If the calculated RoA value of a data request is below this threshold, it is automatically stored off-chain to maintain a historical record of IoT data transactions. Although our approach proposes 7.5 as a threshold value to support on-chain data allocation decisions, it can be changed based on the system administrator and IoT system requirements.

5.6 Performance Evaluation

In this section, we instantiate the data allocation mechanism in two commonly used IoT-blockchain architecture styles for the health-care study. Next, we measure the efficiency of the data allocation mechanism in the two architectures (i.e., blockchain-based fog and cloud) in terms of latency, network usage, and energy consumption.

5.6.1 Evaluation goals

We summarize the motivations for the integration of the data allocation mechanism in blockchain-based fog and cloud architectures as follows:

- Assess the effectiveness of the data allocation mechanism by enabling or disabling it in the two IoT-blockchain architecture styles. It ensures flexibility in the system and satisfies the requirements of end-users and service providers.
- Evaluate the performance of the refined blockchain-based cloud and fog architectures in terms of energy consumption, latency, blockchain size, and network usage. It compares

the performance of the two architecture styles when a large number of requests are generated simultaneously.

5.6.2 Simulation environment

The evaluation of the data allocation mechanism consists of two stages: (1) collect, process, and store the context parameters of each IoT data request using FogBus and (2) simulate the data controller using Matlab to calculate the RoA value and support data allocation.

FogBus is a lightweight real-world blockchain framework that integrates IoT, fog/edge, cloud, and blockchain. Figure 5.7 shows the FogBus-enabled sleep apnea analysis prototype presented in [137].

Figure 5.7: FogBus sleep apnea analysis prototype [137].

The setup of the hardware components and their configuration are given below.

- *IoT device:* Pulse oximeter, 1.5V, Bluetooth 4.0, UFT-8 data encoding.
- *Gateway node:* Oppo A77T smartphone, Android 7.1.1.
- *Master node:* Dell Latitude D630 Laptop, *Intel (R) Core (TM) 2 Duo CPU E6550*

@ *2.33 GHz 2 GB DDR2 RAM*, 32-bit, Windows 7, Apache Server 2.4.34, Java 1.6, MySQL 5.6, .NET 3.5, and Aneka 3.1.

- *Worker node:* Raspberry Pi 3, *ARM Cortex A53 quad-core SoC CPU@ 1.4GHz 1 GB LPDDR2 SDRAM*, Raspian Stretch, Apache Server 2.4.34, Java 1.6, and MySQL 5.6.
- *Cloud:* Microsoft Azure B1s Machine, 1vCPU, 1 GB RAM, 2 GB SSD, Windows Server 2010,.NET 3.5 and Aneka 3.1.

Initially, the oximeter collects timestamp, heart beat, and blood oxygen level for one hour of sleep study and transmits it to the gateways in the data controller tier, which keep an internal list of records. Once the recordings are completed, the data controller receives the oximeter data along with the number of sharing points interested in the data (i.e., doctors, hospitals, laboratories, pharmacies, etc.), and the device accuracy measurement. Here, we use Matlab to design and simulate the data controller with its corresponding input, membership functions, and fuzzy rules. We calculate the RoA value of each data request to realize the data allocation mechanism in the blockchain-based cloud and fog architectures using the FogBus framework.

We define the following concrete metrics to evaluate the efficiency of the data allocation mechanism in blockchain-based fog and cloud architectures.

- *Size of blockchain:* Average size of the blockchain in the broker node and cloud VMs.
- *Average latency:* Data access latency to retrieve data from fog nodes (i.e., broker node and worker nodes) and cloud VM. Since we use the FogBus framework [137] as our simulation environment, it can directly provide the latency measures. In particular, its measured latency refers to the overall system latency (i.e., data processing, network propagation delay, and OS delay).

- *Network usage:* The load on the network when the data allocation mechanism is deployed in the blockchain-based fog and cloud.

- *Energy consumption:* The average energy usage of the broker node for the blockchain-based fog and average energy usage of the Azure VM for the blockchain-based cloud to support the blockchain. In particular, we monitored the energy consumption in the broker node and Azure VM by enabling / disabled the data allocation mechanism by Joulemeter [59], which can estimate the energy consumption of runtime applications.

Table 5.4: Simulation parameters.

Parameters	Values
Analysis task:	
Interval between the creation of consecutive data processing requests	5 seconds
Data recording time per processing requests	3 minutes
Pulse oximeter:	
Pulse oximeter signal length	18 KB
Sensing frequency	2 signals per second
WLAN:	
Download speed	7 MBPS
Upload speed	2 MBPS

5.6.3 Result analysis

We describe the performance results of refined blockchain-based cloud and fog architectures.

Size of blockchain

Figure 5.8 shows the estimated storage size in KB of the blockchain-based (a) cloud and (b) fog architectures, i.e., with / without the implementation of the data allocation mechanism. In general, the size of the blockchain increases linearly when the mechanism is not applied in architecture styles, since all IoT data requests are allocated within the blockchain embedded in fog and cloud without considering its limited storage capacity. On the contrary, the implementation of the data allocation mechanism in the refined IoT-blockchain architecture styles ensures a reduction of around 42% on average in blockchain size. In fact, when 100 data requests are executed on the system, the storage size of a blockchain decreases by around 44% on average and by approximately 42% when 20 data requests are executed on the refined blockchain-based cloud and fog, respectively. From the results, we conclude that the implementation of the data allocation mechanism alleviates the storage capacity of the blockchain, since only data requests with a high RoA value are stored within the cloud and fog architectures based on the blockchain.

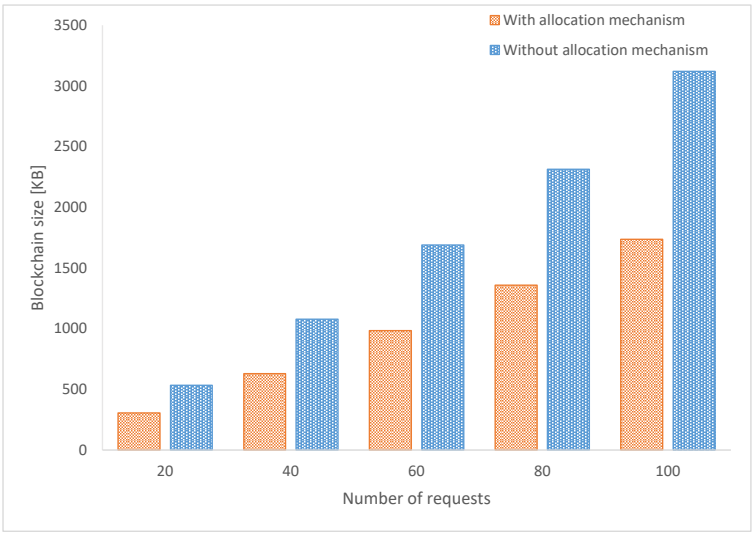

Figure 5.8: Size of the blockchain in KB - with/without the allocation mechanism.

Average latency

Figure 5.9 illustrates the latency of service delivery in seconds when the data allocation mechanism is enabled and disabled in (a) cloud and (b) fog architectures based on blockchain. To simulate the propagation delay from the cloud, we first connect the broker node and the Azure virtual machine through a virtual network of 4 Mbps. In particular, we define the propagation delay in the blockchain-based fog architecture as the delay from the broker node to the worker nodes. Similarly, we measure the propagation delay in the blockchain-based cloud architecture as the delay from the broker node to the cloud virtual machine. The results show that the network propagation delay in the blockchain-based cloud is almost twice as long as in the blockchain-based fog when the data allocation mechanism is not executed. However, its implementation in the blockchain-based cloud contributes to a latency reduction of 36% on average and about 27% in the blockchain-based fog. These results show that the data allocation mechanism effectively reduces the amount of data to be sent to the blockchain, whether it is executed in the blockchain-cloud and fog architectures. Furthermore, since the size of the data chunk to be recorded in the blockchain is not huge, the latency will not differ significantly between the two architectures. Thus, we conclude that the service delivery latency depends mainly on the network propagation delay, which is low in the blockchain-fog architecture since fog nodes are located in a single-hop proximity to where data is collected.

Energy consumption

Figure 5.10 shows the estimated amount of energy consumption in joules when the data allocation mechanism is enabled and disabled in the blockchain-based (a) cloud and (b) fog architectures based on blockchain. Here, the energy consumption in the blockchain-based cloud is approximately 32% more than in the blockchain-based fog when the mechanism is not integrated into the system. These results show that the tasks performed in the cloud

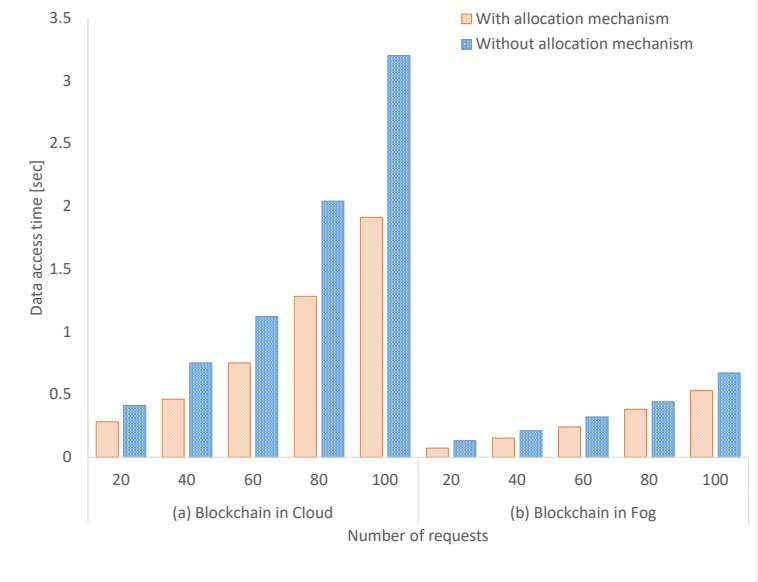

Figure 5.9: Data access time in seconds - with/without the allocation mechanism in (a) blockchain-based cloud and (b) blockchain-based fog.

are complex and require additional computing, storage, and networking resources. In contrast, the implementation of the data allocation mechanism leads to an energy reduction of 28% on average in blockchain-based cloud and fog architectures. When 100 data requests are executed in the system, the energy consumption reaches 410 joules in the refined blockchain-based cloud and just above 285 in the refined blockchain-based fog. In other words, when a high number of data requests are allocated in the blockchain-based fog, the energy consumption is lower compared to the cloud since fog devices are located in single-hop proximity of IoT devices. In contrast, when a high number of data requests are allocated in the blockchain-based cloud, the energy consumption is the same as or higher than in the fog, since cloud servers are located in multi-hop proximity of IoT devices. From these observations, we conclude that the refined blockchain-based fog saves more energy compared to the refined blockchain-based cloud since IoT data requests are processed close to where the data is collected.

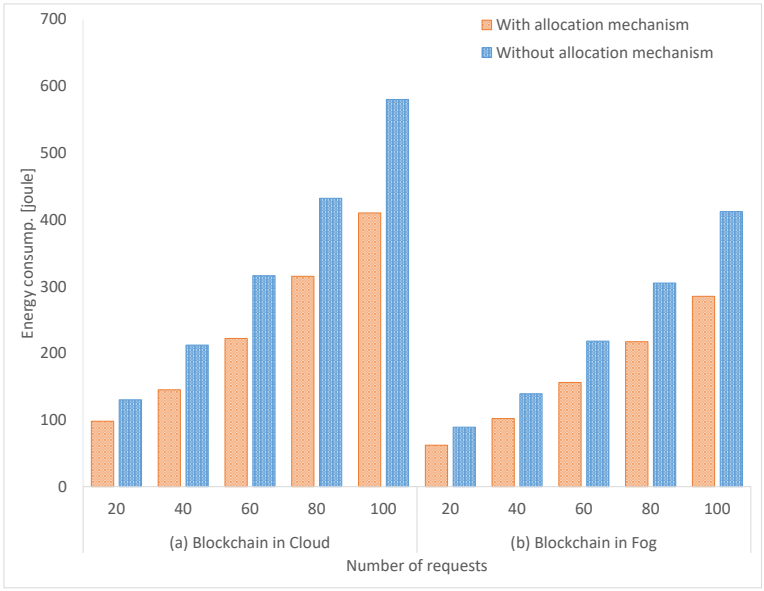

Figure 5.10: Energy consumption in joule - with/without the allocation mechanism in (a) a blockchain-based cloud and (b) a blockchain-based fog.

Network usage

Figure 5.11 illustrates the use of the network in BPS when the data allocation mechanism is enabled and disabled in (a) cloud and (b) fog architectures based on blockchain. The figure shows that the network usage in blockchain-based cloud is about two times more than in the blockchain-based fog when the data allocation mechanism is not integrated into the system. These results show that fog outperforms cloud, as it allows local networking resources to handle IoT data requests. However, integration of the data allocation mechanism reduces network usage in blockchain-based cloud and fog architectures. Approximately 32% of network usage is reduced in the refined blockchain-based fog and just above 24% on average in the refined blockchain-based cloud. In fact, the network usage in refined blockchain-based fog is about 189 BPS, while it reaches a peak over 515 BPS in the refined blockchain-based cloud. Although the implementation of the blockchain in fog and cloud environments increases network usage due to the security mechanisms implemented (e.g., encryption algorithms), blockchain-based fog reports less network usage than blockchain-based cloud, since

fog uses local networking resource which reduces latency and bandwidth consumption in the network.

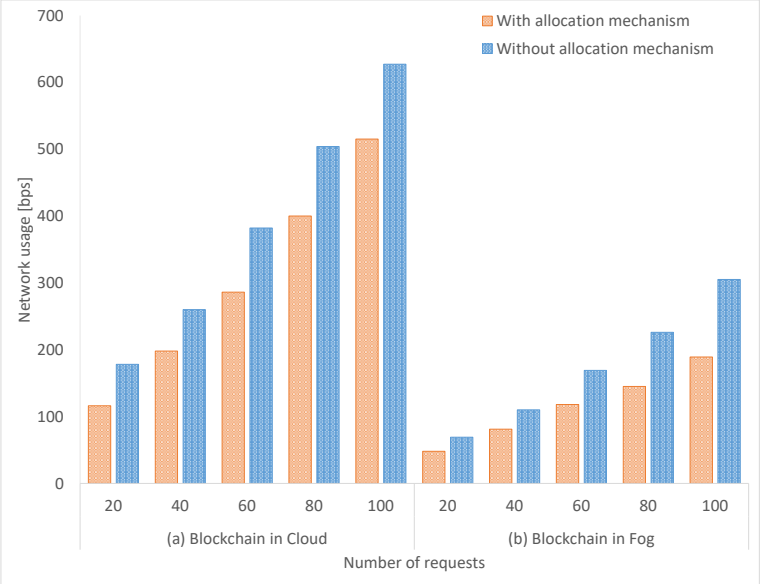

Figure 5.11: Network usage in BPS - with/without the allocation mechanism in (a) blockchain-based cloud and (b) blockchain-based fog.

5.6.4 Performance comparison with alternative decision-making mechanisms

Furthermore, we compare the effectiveness of our approach with the existing alternative decision-making mechanisms for data management. Specifically, we first survey the literature and identify that Logistic Regression [83, 46] and Decision Tree [97, 82, 22] are often considered alternative decision-making approaches. Next, we conduct experiments for comparison of performance with such approaches to show the benefit of our technique. The results of our experiment suggest that our approach incurs a significantly decreasing CPU usage and overall execution time at the broker node, as described in Table 5.5. In particular, the CPU usage in the broker node is achieved at 84% and 51% when running logistic

regression and the decision tree, respectively, while our approach can reduce CPU usage to 30%. Similarly, our approach (0.112s) can outperform all the compared approaches (0.156s, 0.239s) in terms of execution time.

Table 5.5: Fuzzy logic decision-making results compared with state-of-the-art approaches.

Approaches	CPU usage	Execution time
Logistic regression	84%	0.239 secs
Decision tree	51%	0.156 secs
Fuzzy logic	30%	0.112 secs

5.7 Related Work

In this section, we briefly summarize a subset of relevant work for our system.

5.7.1 Fuzzy Logic in IoT

There exist several applications of fuzzy logic in IoT systems, as described below. Vani et al. [139] propose a real-time IoT health monitoring system for elderly people that collects environmental data and uses fuzzy logic to simplify its interpretation and make decisions accordingly. Santamaria et al. [116] propose a fuzzy logic approach that learns customer habits through body sensors and discovers outliers of warning signals to minimize the risk of false alarms. Similarly, Bhunia et al. [14] propose a fuzzy logic-based health care system for a smart city where sensor data is collected (i.e., SPO2, ECG, airflow, temperature, etc.) to support decision-making about true patient conditions, for example weak heart, shock, respiratory problem. In addition to the health care domain, Meana-Llori et al. [91] design a fuzzy logic system that autonomously controls indoor temperature using external climate

conditions that results in a 40% energy reduction. Novilla et al. [96] design a fuzzy logic manufacturing monitoring system that uses temperature and smoke sensors to capture the normal conditions of the manufacturing machines and build a reference model to inform the health status of the machine and provide accurate failure predictions. Mahalle et al. [87] present a fuzzy logic approach to improve trust-based access control in IoT that uses vague values of Experience (EX), Knowledge (KN), and Recommendation (RC) to authorize devices in the IoT network. Another approach presents a fuzzy logic framework to determine the evaluation of employee performance based on IoT data [62]. Globa et al. [44] propose the use of a fuzzy logic mechanism for big data processing in IoT networks to improve performance and reduce the computational costs of complex machine learning algorithms.

5.7.2 Decision-making mechanisms in IoT systems

There exists a considerable body of literature on decision-making mechanisms applied in different domains, such as health-care, manufacturing, and control systems [46]. Lowe et al. [83] propose a logistic regression approach for decision-making about bid/no-bid from contractors in a construction company. Similarly, Young et al. [156] use a regression model to predict diabetes severity index and risk of mortality. Ohno et al. [97] propose the use of a decision tree and fuzzy logic as decision models to select the optimal vaccination strategy. Lopez et al. [82] present a decision model based on a decision tree algorithm that combines relevant health-care criteria for the screening and diagnosis. Furthermore, Chern et al. [22] propose a decision tree model that provides optimal telehealth services and reduces resource misuse. Kara et al. [61] present a diagnostic disease system based on artificial neural networks that collect data from small mobile devices. Similarly, Burke et al. [17] rely on artificial neural networks to improve the accuracy of cancer prediction. In addition, Ting et al. [135] propose a diagnostic system for obstructive sleep apnea based on decision

tree algorithms, which can perform automatic feature selection. Timus et al. [134] present a classifier K-Nearest-Neighbor (k-NN) to determine the types of sleep apnea. A fuzzy decision tree is proposed for classification and prediction problems [9, 36].

Our proposed approach differs from the aforementioned works, since we have considered the use of fuzzy logic to derive on-chain allocation decisions based on multiple context parameters. Specifically, we design a data controller based on fuzzy logic that handles multiple context parameters, e.g., data, network, and quality, to calculate the RoA value of each IoT data request and support on-chain data allocation. The RoA value is used as a threshold measurement to decide which data request needs to be allocated within the blockchain or stored off-chain, for example, cloud database. Moreover, the realization of the data allocation mechanism in the two IoT-blockchain architecture styles leads to refinements which are analyzed from an abstract level by proposing a four-tier abstraction, i.e., the IoT tier, the data controller tier, the fog tier and the cloud tier. To demonstrate the effectiveness of our approach, we instantiate it in blockchain-based cloud and fog architectures and evaluate their performance in terms of network usage, latency, energy consumption, and blockchain storage.

5.8 Discussion

We have shown the implementation of the data allocation mechanism in two commonly used architectures in IoT implementations that integrate blockchain in IoT systems, i.e., blockchain-based cloud and fog [114, 77]. Although we have chosen these architecture styles as a way to illustrate our approach, it can continue to work in other styles. In particular, an alternative data management strategy could be to have the blockchain as a separate network in fog or cloud environments where each block stores only the hash of the data and data

address of the relevant data while maintaining the raw-data in the cloud to meet IoT system requirements. Moreover, we can develop a market-based mechanism to decide the utility improvement of using fog and cloud environments or secure platforms (i.e., blockchain-based cloud and fog) for IoT data allocation considering the cost, QoS requirements and constraints imposed by each hosting environment. In this model, users or service providers can be charged based on pay-as-you-go or subscription fee to decide when to use the normal fog and cloud or a secure environments (i.e., blockchain-based cloud and fog) for storing IoT data.

5.9 Conclusion

In this study, we identify some architecture significant requirements for developing a data-driven approach that supports data allocation in IoT systems supported with blockchain. To meet these requirements, we propose a novel data allocation mechanism that calculates the RoA value of each IoT data request based on multiple context parameters to decide its on-chain allocation. The mechanism relies on the design of a data controller based on fuzzy logic that extracts context parameters of each data request, e.g., data, network, and quality to determine the RoA value which is used as a threshold measurement to decide which data request needs to be stored within the blockchain or allocated off-chain, i.e., cloud database. Moreover, we enrich the two commonly used IoT-blockchain architecture styles supported by fog and cloud with the data allocation mechanism, where we introduce a data controller tier between the IoT tier and the fog tier to handle on-chain allocation decisions in real-time. To evaluate the effectiveness of our approach, we instantiate the blockchain-based cloud and fog architectures in a health-care example using FogBus framework. We conducted several experiments to measure latency, energy consumption, network usage, and blockchain storage in refined architecture styles. The performance evaluation suggests that latency is reduced by 36% in the refined blockchain-based cloud and about 27% in the refined

blockchain-based fog. Similarly, energy consumption is reduced on average by 28% in the refined blockchain-based cloud and fog. Furthermore, network usage is reduced by 32% in the refined blockchain-based fog and 24% in the refined blockchain-based cloud.

Furthermore, our objective is to investigate the generality of the proposed mechanism and its application to other alternative styles that take advantage of private and public blockchains. We also aim to evaluate the performance of the refined IoT-blockchain architecture styles in a real environment by integrating other sensors (e.g., temperature and air quality) to improve the calculation of the RoA value to minimize the risk of uncertainty in data allocation decisions.

Chapter Six

Reflection and Appraisal

This chapter revisits the research questions proposed in Chapter 1 and includes a reflection on the evaluation performed in each contribution.

6.1 Analysis of the Research Questions

This section discusses to what extent the four research questions presented in this thesis have been addressed.

- **RQ1:** What software quality attribute requirements, architectural tradeoffs, and design decisions are commonly discussed for the architectural design of IoT systems supported by blockchain?

  In chapter 2, we conducted a systematic literature review (SLR) [152] to investigate the common quality attributes, architectural tradeoffs, and design decisions for IoT systems supported by blockchain reported in the primary studies. This body of architectural knowledge can help software architects and developers reflect and reason about design decisions to achieve particular quality attributes and tradeoffs. A total of 100

primary studies were identified, evaluated, and analyzed, leading to the identification of security, scalability, performance, and interoperability as common quality attributes to consider in the design process of this category of systems. Additionally, we extracted common architectural design decisions that include data distribution and computation, blockchain location, type of blockchain, consensus protocol, and blockchain data structure.

The results of the SLR also allowed us to identify gaps and opportunities for research in (i) identification of other quality attributes that are relevant to operational IoT systems supported by blockchain, such as mobility and adaptability, (ii) analysis of design decisions at system level architecture, (iii) evaluation of the real-world impact of architectural design decisions, and (iv) trade-offs among the quality attributes and identified design decisions, (v) identification of architectural tactics, (vi) refinements of existing architectural styles based on the identified architectural design decisions, and (vii) mechanism for data and computation distribution. From the aforementioned gaps in the literature, this work attempted to address the last three to contribute to the development of IoT systems supported by blockchain in academia and the industry. This chapter was derived in part from [152].

- **RQ2:** What architectural tactics can be documented from identified architectural design decisions to build candidate architectures for IoT systems supported by blockchain that achieve particular quality attribute requirements?

  In chapter 3, we codified a catalog of architectural tactics derived from the architectural design decisions identified in the primary studies. The tactics can guide software architects and developers in the architectural design of a candidate architecture that meets the desired qualities of the system. To this end, we selected the architectural design decisions commonly identified in primary studies to satisfy particular quality attribute goals [152]. The analysis of design decisions led us to the identification of 12

architectural tactics, which were extracted from the literature based on (i) explicitly stated quality attributes, (ii) inferred quality attributes from the literature, and (iii) commonly reported components and their relations.

In particular, we identified tactics for security, scalability, performance, and interoperability; however, there were other qualities of the system, such as mobility and adaptability, that were not considered by the primary studies of IoT systems supported by blockchain. The results also revealed other gaps and opportunities for research and development as follows: (i) investigation is required to evaluate the impact of the architectural tactics in this category of systems and (ii) additional research is needed to explore the trade-offs among the quality attributes and identified tactics. These ideas for future research can require extensive collaboration between industry and academia to implement, deploy, and evaluate architectural tactics and control quality attributes in large-scale IoT systems supported by blockchain. Finally, we used the design-science-based evaluation approach to reflect on the catalog of architectural tactics. This chapter was derived in part from the work presented in [152].

- **RQ3:** What reference architecture styles can be implied to guide the development of IoT systems supported by the blockchain? How to assess the fitness of the reference architectures with respect to particular system qualities? What are the applications and usage domains that can benefit from the reference architectures?

  In chapter 4, we codified a set of reference architecture styles by inspecting some representative examples in the literature. In particular, commonly used styles were selected to argue about design decisions and tradeoffs that can guide software architects and developers when building IoT systems supported by blockchain. Many architectural design decisions must be made when designing this category of systems. However, current developments were still designed impromptu due to the lack of systematic analysis of architectural design issues of blockchain and IoT technologies.

We used the Architectural Trade-off Analysis Method (ATAM), a systematic procedure, to understand the implicit tradeoff architectures and guide the development of IoT systems supported by blockchain. Specifically, the ATAM was used to assess the general fitness of the identified architectural styles regarding the quality attributes promoted by this category of systems. The results of the ATAM analysis led to refinements of existing architectures, known as variants. We complemented the qualitative evaluation performed by ATAM with simulation to not only reveal how the identified architectural styles meet the quality attributes, but also provide insight into the trade-offs among quality goals. This chapter was partially derived from the work presented in [154].

- **RQ4:** What are the design decisions driving the development of a dynamic data allocation mechanism for IoT systems supported by blockchain? How can a data allocation mechanism be effectively engineered in IoT systems supported by blockchain, considering context information, quality attributes, IoT constraints, and inherent limitations of blockchain? How can the reference architecture styles and variants be enriched with a data allocation mechanism to decide on on-chain and off-chain storage?

In chapter 5, we developed a data allocation mechanism for IoT systems supported by blockchain to decide on on-chain and off-chain storage [153]. The mechanism relied on a data controller supported by fuzzy logic and context information to decide on which data should be recorded on the blockchain (i.e., on-chain) or external storage (i.e., off-chain). Specifically, we recorded context information from the IoT environment (e.g., data, network, and quality) to calculate the Rating of Allocation (RoA) value of each data request, which is used as a threshold value for allocation decisions. Furthermore, we illustrated how the design and realization of the mechanism lead to refinements of two commonly used IoT-blockchain architectural styles (i.e., blockchain-based cloud and fog).

Our approach handled IoT constraints and inherent blockchain limitations, which

could result in incompleteness, inconsistency, imprecision, and / or vagueness of the data, with the introduction of a mechanism that takes advantage of context awareness and fuzzy logic. We did not investigate how contextual information is extracted from the system; instead, we assumed that the system provides a controller to capture context information and decide on data allocation. Context-aware computing and fuzzy logic were not new paradigms in IoT; however, most data management and allocation approaches proposed the use of machine learning algorithms for decision-making, which could result in heavy computation. This chapter was derived from the work presented in [153].

6.2 Reflection on the Research

This section reflects on the approach and evaluation proposed in this thesis in terms of the design aspects of the simulation environment, including overhead and scalability.

6.2.1 Simulation Environment

In chapter 4, we implemented a simulation environment to implement the architectural styles and their refinements. We set up an Ethereum blockchain with Proof-of-Authority (PoA) as consensus protocol and deployed gateways and IoT devices as virtualized components using Virtual Box. We then performed a simulation of the styles under different configurations by varying the number of transactions to be sent to the blockchain.

In chapter 5, we used FogBus [138], a lightweight real-world blockchain framework that integrates IoT, edge, cloud, and blockchain to evaluate the proposed data allocation mechanism. The framework facilitates the deployment of IoT applications and multiple computer instances, as well as implements authentication and encryption techniques to protect

sensitive data in the blockchain network. In addition to these features, the framework allowed us to implement the fuzzy logic controller at the edge to decide on which data generated by IoT devices need to be recorded on the blockchain or in external storage.

Although the use of a controlled environment instead of a real IoT-blockchain implementation can be debatable, the evaluation of the proposed approaches through simulation enabled us to conduct repeatable and free-of-cost experiments. Furthermore, the simulation allowed us to deploy scenarios under different settings and make an abstraction of low-level details related to a real testbed. However, more research is needed to evaluate the effectiveness of our approaches in a real data-centric IoT system supported by blockchain.

6.2.2 Computational Overhead

The performed experiments to demonstrate the effectiveness of our approaches may carry out some hidden computational overhead due to the number of IoT devices and blockchain nodes in the network. In particular, the overhead observed in the simulation is similar to the one experienced in the physical infrastructure and comes mainly from two sources: (i) the transmission of the data from the IoT devices to the blockchain, and (ii) the end-to-end latency in the blockchain network.

6.2.3 Dealing with IoT and Blockchain Dynamics

In self-adaptive software systems, dynamics refers to changing conditions in the environment, which lead to continuous adaptations in the system to satisfy the quality attributes of interest [55]. Regarding IoT systems supported by blockchain, IoT systems are data-driven in nature, characterized by high velocity, high volume of data, and high mobility, making securing data and its management a challenge. However, blockchain provides a decentralized environment

and attractive features (e.g., immutability, transparency, and auditability) for distributed and secure IoT data management; however, it also poses some technical restrictions, such as limited computing power and data storage. These characteristics can lead to unpredictable networks, where data management and its allocation become a significant issue [147]. Our experimental environment emulates a blockchain-enabled IoT system; in which the main source of dynamics comes from the IoT network, where each device collects an unpredictable amount of data from the environment and connects and disconnects from the network-based computational resources.

Chapter Seven

Conclusion Remarks and Future Work

This chapter summarizes our contributions related to the research questions and discusses potential future directions derived from the findings in this thesis.

7.1 Contributions

The goal of this thesis is to provide guidance for software architectures and investigate a new mechanism for data management and its allocation for IoT that facilitate the development of IoT systems supported by blockchain. In particular, this thesis makes the following contributions.

- **A systematic literature review (SLR) of architectural design decisions in IoT systems supported by blockchain.** In current literature, there are inadequacies in a disciplined understanding of the software quality attributes and the tradeoffs that can drive the development of IoT systems through blockchain. To address this issue, we critically examine 100 primary studies on the integration of blockchain and IoT to identify common quality attributes, architectural tradeoffs, and design deci-

sions that can serve as primary drivers when architecting IoT systems supported by blockchain. In particular, the results of the SLR led us to the identification of gaps and opportunities for further research related to inadequacies in the analysis of other quality attributes that are relevant to operational IoT systems supported by blockchain and the identification of architectural tactics and styles that support this category of systems.

- **A catalog of architectural tactics for IoT systems supported by the blockchain.** Many IoT systems supported by blockchain are still designed ad hoc, due to the general absence of a comprehensive body of architectural knowledge that systematically investigates and documents the design decisions that drive the development of this category of systems. To overcome this issue, we present a catalog of architectural tactics for IoT systems supported by blockchain, derived from the commonly identified design decisions in the primary studies [152]. The main goal of the tactics is to provide software architects and developers with a set of architectural design options to build candidate architectures for IoT systems supported by blockchain that fulfill the desired quality attributes. In particular, we identified a total of 12 tactics that promote specific quality attributes such as security, scalability, performance, and interoperability. The goal of the tactics is to guide architects and developers in designing IoT systems supported by blockchain to satisfy specific quality attribute goals.

- **A set of reference architecture styles and variants for building IoT systems supported by blockchain**. The majority of IoT systems supported by blockchain have been designed without reflecting and reasoning on the underlying styles that support them and providing clear guidelines for building this category of systems. To address this issue, we extracted common architecture styles from the primary studies and analyzed them using ATAM to assess their general fitness with respect to the desired quality attributes. The results of the ATAM analysis have led to refinements

of existing architectures, known as variants. Both styles and variants are documents that use the architectural pattern language to ease their adoption and provide software architects and designers with a set of architectural design options that meet the qualities of the system. We complement the ATAM results with a quantitative analysis that evaluates the applicability and efficiency of the styles and their variants using simulation.

- **A data allocation mechanism for IoT systems supported by blockchain.** Several approaches have been introduced to develop IoT systems supported by blockchain, without considering the IoT constraints and inherent limitations of blockchain [147]. Furthermore, the limited computation and data storage space in public blockchains is also not considered a key requirement when designing this category of systems [151]. To reason about design decisions in light of constraints in IoT and blockchain and to provide guidelines on the architectural design of this category of systems, we implemented a fuzzy logic controller at the edge of the network to extract context information from sensor data (e.g., data, network, and quality) and calculate the Rate of Allocation (RoA) value [153]. This value is used as a threshold to decide which data should be recorded on the blockchain or external storage. The design and realization of the mechanism led to refinements of two existing architecture styles, which were evaluated in terms of performance using FogBus.

7.2 Future Directions

This section concludes the thesis by summarizing some ideas for future research on software architectures and design decisions for IoT systems supported by blockchain that have been discussed in previous chapters and in other new directions.

7.2.1 Extension of the Catalog of Tactics

The findings in chapters 2 and 3 present some opportunities to reflect on the proposed catalog of architectural tactics and to analyze its extension in different ways.

- Consistent with the findings in Chapter 3, some architectural tactics present variations, such as side chain and caching offload, that could be annotated in the catalog using the pattern language. Other tactics present examples that respond to a general implementation of the core tactic and whose implementation could suggest small variations. Thus, it would be of great value to software architects and developers to extend the catalog of tactics of IoT systems supported by blockchain to add variations that influence the achievement of particular system qualities.

- Consistent with the findings in Chapter 2, the SLR results show that there are some quality attributes, such as mobility and adaptability, that do not have architectural support in the literature. As tactics are design decisions to satisfy particular qualities of the system, we propose to extend the proposed catalog of tactics to add other tactics that address the identified quality attributes. By adding these tactics, we will provide full coverage of the desired system qualities and yield candidate architectures that can guide software architects and developers in building this category of systems.

- The results in Chapter 3 revealed that the proposed catalog of tactics is not conclusive; instead, new tactics can be added to meet all the desired quality attributes promoted by IoT systems supported by blockchain. For instance, our catalog of tactics has focused on security, scalability, performance, and interoperability and mainly serves to realize them in the operating environment. However, we envision that new variants can still emerge when other qualities can be considered. For example, considering mobility and adaptability as architectural concerns can lead to new variants of architectural tactics.

These variants can be different in the way data is handled (e.g., on-chain or off-chain), and deployed (e.g., at which time and context) in the architectural design process.

7.2.2 Analysis of the Architecture Styles and their variants

The qualitative and quantitative analysis of architecture styles and their variants for IoT systems supported by blockchain presented a number of opportunities for future research, as follows:

- In chapter 4, we partially use ATAM to understand the tradeoffs of the reference architecture styles and support architects and designers i in the choice of an architecture for underlying IoT systems supported by blockchain. Additionally, we obtain the quality attributes and scenarios from a healthcare case study, rather than from stakeholders.
- Analyzing three architecture styles for IoT systems supported in blockchain identified in primary studies limits the generality of ATAM evaluation. Other representative styles could be included to build software architectures and provide software architectures with many candidate architectures that meet the desired system qualities.
- Consistent with the findings in Chapter 4, other architectural evaluation methods, such as the Cost Benefit Analysis Method (CBAM), can be used to complement the ATAM results. In particular, the scenarios and architectural strategies defined in ATAM can be used to model in terms of time and cost.
- The quantitative evaluation of the architecture styles and their variants was carried out through simulation in a controlled environment, which enables faster experimentation but at the same time limits the generality of the results. As a result, experiments should be performed in a real environment with numerous IoT devices to get different results based on the implemented styles.

7.2.3 Proactive Data Allocation Mechanism

We propose a novel data allocation mechanism based on context information and fuzzy logic, which decides how to efficiently allocate IoT data in the blockchain in light of IoT constraints and inherent limitations of the blockchain [147]. This mechanism also limits the usage of computation and storage on the blockchain by making appropriate decisions about which data needs to be recorded on-chain or off-chain. However, the use of fuzzy logic can lead to incorrect allocation decisions due to inaccurate data generated by IoT devices (e.g., incomplete, imprecise, and missing information). In addition, the fuzzy rules defined in the inference engine are completely dependent on human knowledge and expertise, as well as need to be updated based on the operational context. As a result, machine learning and/or neural network approaches can be used to profile allocation decisions in this category of systems and improve their efficiency. By leveraging on machine learning and/or neural networks, the data controller should be able to learn from the IoT constraints and inherent limitations of blockchain, as well as from the accumulated knowledge that can be used to improve allocation decisions.

Another direction is the use of AI (Artificial Intelligence) explanations to understand the allocation decisions made by the controller. In particular, the AI explanation is used mainly for classification and regression tasks and determines how much each feature in the system has contributed to the given output. Thus, when we request explanations for a given decision, it will come along with the features that contribute to the output.

7.3 Conclusion Remarks

Blockchain offers a distributed ledger, in which IoT data can be recorded as immutable transactions and processed in consensus by some blockchain nodes [147]. Due to its at-

tractive features, such as transparency, traceability, auditability, and accountability, the blockchain is being considered as the backbone architecture of distributed and secure IoT data management [89]. However, the integration of blockchain and IoT is still facing some challenges. IoT systems are data-driven in nature, characterized by high velocity, high volume of data, and high mobility, making data security and its management a challenge. However, blockchain provides a decentralized environment and attractive features (e.g., immutability, transparency, and auditability) for distributed and secure IoT data management; however, it also poses some technical restrictions, such as limited computing power and data storage. Thus, developing a mechanism for data management and its allocation that deals with the constraints of IoT and technical limitations of blockchain would be of great value for architecting IoT systems supported by blockchain.

Although there are several approaches to the application of blockchain in IoT systems [111, 89, 147], only a few attempts have been identified in the literature that examine the integration of both technologies from the perspective of software architecture. As these technologies become more prevalent due to the inherent benefits of the blockchain combined distributed IoT data management, a need will arise for guidance on software architectures and design decisions for the development of IoT systems supported by blockchain to meet the desired quality attributes.

This thesis provides architectural knowledge and systematic guidelines for the development of IoT systems supported by blockchain. First, this thesis examines common quality attributes, architectural tradeoffs, and design decisions for IoT systems supported by blockchain in the primary studies identified in the systematic literature review (SLR). Second, it investigates the architectural tactics that can be derived from the architectural design decisions identified in the primary studies. These tactics can guide architects and developers in the architectural design process of software architectures for IoT systems supported by blockchain to satisfy their intended quality attributes. Third, this thesis identifies reference

architecture styles for IoT systems supported by blockchain using representative examples from the literature. We use ATAM analysis to understand the tradeoff points of the reference architectures, which result in refinements of them. Finally, this thesis explores the design of a data allocation mechanism for IoT systems supported by blockchain to decide on on-chain and off-chain storage and deals with IoT constraints and blockchain technical limitations. The goal is to help software architects and developers extend their architectural design reasoning on the software architectures and data allocation mechanism toward the development of IoT systems supported by blockchain to achieve the desired quality attributes.

Appendix One

Appendix 1

A.1 List of Primary Studies

Table A.1 presents the list of the 100 primary studies.

Table A.1: Appendix 1

ID	Title	Short name	Author(s)	Year
PS1	Towards blockchain-based intelligent transportation systems	Blockchain Transportation	Yuan, Yong and Wang, Fei-Yue	2016
PS2	Towards an optimized blockchain for IoT	Optimized blockchain	Dorri, Ali and Kanhere, Salil S and Jurdak, Raja	2017
PS3	Block-VN: A Distributed Blockchain Based Vehicular Network Architecture in Smart City	Block-VN	Sharma, Pradip Kumar and Moon, Seo Yeon and Park, Jong Hyuk	2017

Continued on next page

Table A.1 – *Continued from previous page*

ID	Title	Short name	Author(s)	Year
PS4	MeDShare: Trust-Less Medical Data Sharing among Cloud Service Providers via Blockchain Using Blockchains to Strengthen the Security of IoT	MeDShare	Xia, QI and Sifah, Emmanuel Boateng and Asamoah, Kwame Omono and Gao, Jianbin and Du, Xiaojiang and Guizani, Mohsen	2017
PS5	A Software Defined Fog Node Based Distributed Blockchain Cloud Architecture for IoT	SDN-Blockchain	Sharma, Pradip Kumar and Chen, Mu-Yen and Park, Jong Hyuk	2017
PS6	Blockchain Based Distributed Control System for Edge Computing	Blockchain for Edge	Stanciu, Alexandru	2017
PS7	Towards better availability and accountability for IoT updates by means of a blockchain	IoT Updates	Boudguiga, Aymen and Bouzerna, Nabil and Granboulan, Louis and Olivereau, Alexis and Quesnel, Flavien and Roger, Anthony and Sirdey, Renaud	2017

Continued on next page

Table A.1 – *Continued from previous page*

ID	Title	Short name	Author(s)	Year
PS8	Towards blockchain-based auditable storage and sharing of IoT data	Blockchain auditable storage	Shafagh, Hossein and Burkhalter, Lukas and Hithnawi, Anwar and Duquennoy, Simon	2017
PS9	Integrating blockchain for data sharing and collaboration in mobile healthcare applications	Blockchain for data sharing	Liang, Xueping and Zhao, Juan and Shetty, Sachin and Liu, Jihong and Li, Danyi	2018
PS10	Peer to peer for privacy and decentralization in the internet of things	P2P privacy in IoT	Conoscenti, Marco and Vetro, Antonio and De Martin, Juan Carlos	2017
PS11	Vegvisir: A Partition-Tolerant Blockchain for the Internet-of-Things	Vegvisir	Karlsson, Kolbeinn and Jiang, Weitao and Wicker, Stephen and Adams, Danny and Ma, Edwin and van Renesse, Robbert andWeatherspoon, Hakim	2018
PS12	Blockchain based hybrid network architecture for the smart city	Hybrid BC-IoT	Sharma, Pradip Kumar and Park, Jong Hyuk	2018

Continued on next page

Table A.1 – *Continued from previous page*

ID	Title	Short name	Author(s)	Year
PS13	An Out-of-band Authentication Scheme for Internet of Things Using Blockchain Technology	IoT authentication	Wu, Longfei and Du, Xiaojiang and Wang, Wei and Lin, Bin	2018
PS14	Towards using blockchain technology for IoT data access protection	IoT protection-blockchain	Rifi, Nabil and Rachkidi, Elie and Agoulmine, Nazim and Taher, Nada Chendeb	2018
PS15	On design issues and architectural styles for blockchain-driven IoT services	Architectural styles	Liao, Chun-Feng and Bao, Sheng-Wen and Cheng, Ching-Ju and Chen, Kung	2017
PS16	IoTChain: A blockchain security architecture for the Internet of Things	IoTChain	Alphand, Olivier and Amoretti, Michele and Claeys, Timothy and Dall'Asta, Simone and Duda, Andrzej and Ferrari, Gianluigi and Rousseau, Franck and Tourancheau, Bernard and Veltri, Luca and Zanichelli, Francesco	2018
PS17	Blockchain as a Service for IoT	Blockchain as a Service for IoT	Samaniego, Mayra and Deters, Ralph	2016

Continued on next page

Table A.1 – *Continued from previous page*

ID	Title	Short name	Author(s)	Year
PS18	Towards data assurance and resilience in IoT using blockchain	IoT data assurance	Liang, Xueping and Zhao, Juan and Shetty, Sachin and Li, Danyi	2017
PS19	Blockchain platform for industrial internet of things	BC-IIoT	Bahga, Arshdeep and Madisetti, Vijay K	2016
PS20	A decentralized solution for IoT data trusted exchange based-on blockchain	IoT exchange	Huang, Zhiqing and Su, Xiongye and Zhang, Yanxin and Shi, Changxue and Zhang, Hanchen and Xie, Luyang	2017
PS21	Adaptable blockchain-based systems: A case study for product traceability	Adaptabe blockchain	Lu, Qinghua and Xu, Xiwei	2017
PS22	An Approach to Data Privacy in Smart Home using Blockchain Technology	Privacy SH	Dang, Thanh Long Nhat and Nguyen, Minh Son	2018
PS23	BlAsT: Blockchain-Assisted Key Transparency for Device Authentication	BlAsT	Gattolin, Alessandro and Rottondi, Cristina and Verticale, Giacomo	2018

Continued on next page

Table A.1 – *Continued from previous page*

ID	Title	Short name	Author(s)	Year
PS24	IoT data integrity verification for cyber-physical systems using blockchain	Integrity CPS	Machado, Caciano and Fröhlich, Antônio Augusto Medeiros	2018
PS25	An architecture pattern for trusted orchestration in IoT edge clouds	Pahl, Claus and El Ioini, Nabil and Helmer, Sven and Lee, Brian	2018	
PS26	A dynamic scalable blockchain based communication architecture for IoT	Scalable blockchain for IoT	Qiu, Han and Qiu, Meikang and Memmi, Gerard and Ming, Zhong and Liu, Meiqin	2018
PS27	Approaches to Front-End IoT Application Development for the Ethereum Blockchain	Front-End IoT Dev	Pustišek, Matevž and Kos, Andrej	2018
PS28	A Peer-to-Peer Architecture for Distributed Data Monetization in Fog Computing Scenarios	P2P Data Monetization	de la Vega, Francisco and Soriano, Javier and Jimenez, Miguel and Lizcano, David	2018

Continued on next page

Table A.1 – *Continued from previous page*

ID	Title	Short name	Author(s)	Year
PS29	Blockchain-Based Internet of Vehicles: Distributed Network Architecture and Performance Analysis	Blockchain-based IoV	Jiang, Tigang and Fang, Hua and Wang, Honggang	2019
PS30	Delay and Communication Tradeoffs for Blockchain Systems With Lightweight IoT Clients	Blockchain Lightweight IoT clients	Danzi, Pietro and Kalør, Anders E and Stefanović, Čedomir and Popovski, Petar	2019
PS31	BIFF: A Blockchain-based IoT Forensics Framework with Identity Privacy	BIFF	Le, Duc-Phong and Meng, Huasong and Su, Le and Yeo, Sze Ling and Thing, Vrizlynn	2019
PS32	Integration of Fog Computing and Blockchain Technology Using the Plasma Framework	Fog and blockchain using Plasma	Ziegler, Michael Herbert and Groβmann, Marcel and Krieger, Udo R	2019
PS33	Emergency Service for Smart Home System Using Ethereum Blockchain: System and Architecture	Emergency SH	Tantidham, Thitinan and Aung, Yu Nandar	2019

Continued on next page

Table A.1 – *Continued from previous page*

ID	Title	Short name	Author(s)	Year
PS34	New Blockchain-Based Architecture for Service Interoperations in Internet of Things	Interoperability IoT	Viriyasitavat, Wattana and Da Xu, Li and Bi, Zhuming and Sapsomboon, Assadaporn	2019
PS35	IoT Meets Blockchain: Parallel Distributed Architecture for Data Storage and Sharing	IoT Meets Blockchain	Liu, Shaowei and Wu, Jing and Long, Chengnian	2018
PS36	An Efficient Forensics Architecture in Software-Defined Networking-IoT Using Blockchain Technology	Forensic SDN	Pourvahab, Mehran and Ekbatanifard, Gholamhossein	2019
PS37	Privacy Improvement Architecture for IoT	Privacy IoT	Addo, Ivor D and Ahamed, Sheikh I and Yau, Stephen S and Buduru, Arun	2018
PS38	Blockchain and IoT Data Analytics for Fine-Grained Transportation Insurance	Blockchain transport insurance	Li, Zengxiang and Xiao, Zhe and Xu, Quanqing and Sotthiwat, Ekanut and Goh, Rick Siow Mong and Liang, Xueping	2018

Continued on next page

Table A.1 – *Continued from previous page*

ID	Title	Short name	Author(s)	Year
PS39	Blockchain-based Ownership Management for Medical IoT (MIoT) Devices	MIoT	Alblooshi, Mansoor and Salah, Khaled and Alhammadi, Y	2019
PS40	A Two-Layer-Consensus Based Blockchain Architecture for IoT	Two-layer consensus	Bai, He and Xia, Geming and Fu, Shaojing	2019
PS41	Maximizing the System Energy Efficiency in the Blockchain Based Internet of Things	Energy blockchain	Fu, Shu and Zhao, Lian and Ling, Xinhua and Zhang, Haijun	2019
PS42	A Hierarchical Sharding Protocol for Multi-Domain IoT Blockchains	Sharding	Tong, Wei and Dong, Xuewen and Shen, Yulong and Jiang, Xiaohong	2019
PS43	Hybrid-iot: Hybrid blockchain architecture for internet of things-pow sub-blockchains	Hybrid-IoT	Sagirlar, Gokhan and Carminati, Barbara and Ferrari, Elena and Sheehan, John D and Ragnoli, Emanuele	2016
PS44	Management and monitoring of IoT devices using blockchain	Management IoT	Košt'ál, Kristián and Helebrandt, Pavol and Belluš, Matej and Ries, Michal and Kotuliak, Ivan	2019

Continued on next page

Table A.1 – *Continued from previous page*

ID	Title	Short name	Author(s)	Year
PS45	Fog Computing Architecture Based Blockchain for Industrial IoT	Fog IIoT	Jang, Su-Hwan and Guejong, Jo and Jeong, Jongpil and Sangmin, Bae	2019
PS46	Blockchain-based secure firmware management system in IoT environment	Blockchain firmware IoT	Son, Minsung and Kim, Heeyoul	2019
PS47	Privacy-preserving blockchain based IoT ecosystem using attribute-based encryption	Privacy-preserving blockchain	Rahulamathavan, Yogachandran and Phan, Raphael C-W and Rajarajan, Muttukrishnan and Misra, Sudip and Kondoz, Ahmet	2017
PS48	An Efficient and Compacted DAG-based Blockchain Protocol for Industrial Internet of Things	DAG	Cui, Laizhong and Yang, Shu and Chen, Ziteng and Pan, Yi and Xu, Mingwei and Xu, Ke	2019
PS49	Opportunistic Mobile IoT with Blockchain Based Collaboration	Opportunistic IoT	Chamarajnagar, Ravishankar and Ashok, Ashwin	2018

Continued on next page

Table A.1 – *Continued from previous page*

ID	Title	Short name	Author(s)	Year
PS50	MediChainTM: A Secure Decentralized Medical Data Asset Management System	MediChainTM	Rouhani, Sara and Butterworth, Luke and Simmons, Adam D and Humphery, Darryl G and Deters, Ralph	2018
PS51	Homomorphic Consortium Blockchain for Smart Home System Sensitive Data Privacy Preserving	Homomorphic blockchain	She, Wei and Gu, Zhi-Hao and Lyu, Xu-Kang and Liu, Qi and Tian, Zhao and Liu, Wei	2019
PS52	A Blockchain-Based Decentralized Security Architecture for IoT	Decentralized architecture for IoT	Angin, Pelin and Mert, Melih Burak and Mete, Okan and Ramazanli, Azer and Sarica, Kaan and Gungoren, Bora	2018
PS53	Analysis of the Communication Traffic for Blockchain Synchronization of IoT Devices	Traffic for blockchain	Danzi, Pietro and Kalor, Anders Ellersgaard and Stefanovic, Cedomir and Popovski, Petar	2018

Continued on next page

Table A.1 – *Continued from previous page*

ID	Title	Short name	Author(s)	Year
PS54	Using Blockchains to Strengthen the Security of IoT	Strengthen IoT Security	Kouzinopoulos, Charalampos S and Spathoulas, Georgios and Giannoutakis, Konstantinos M and Votis, Konstantinos and Pandey, Pankaj and Tzovaras, Dimitrios and Katsikas, Sokratis K and Collen, Anastasija and Nijdam, Niels A	2018
PS55	Decentralized On-Demand Energy Supply for Blockchain in Internet of Things: A Microgrids Approach	Energy for blockchain	Li, Jianan and Zhou, Zhenyu and Wu, Jun and Li, Jianhua and Mumtaz, Shahid and Lin, Xi and Gacanin, Haris and Alotaibi, Sattam	2019
PS56	Blockchain Based Authentication and Authorization Framework for Remote Collaboration Systems	Blockchain-based Authentication	Widick, Logan and Ranasinghe, Ishan and Dantu, Ram and Jonnada, Srikanth	2019

Continued on next page

Table A.1 – *Continued from previous page*

ID	Title	Short name	Author(s)	Year
PS57	Design and implementation of an automated and decentralized pollution monitoring system with blockchains, smart contracts, and LoRaWAN	Design IoT	Niya, Sina Rafati and Jha, Sanjiv S and Bocek, Thomas and Stiller, Burkhard	2018
PS58	Chained of Things: A Secure and Dependable Design of Autonomous Vehicle Services	Chained of Things	Hasan, Md Golam Moula Mehedi and Datta, Amarjit and Rahman, Mohammad Ashiqur and Shahriar, Hossain	2018
PS59	Blockchain and IoT-Based Cognitive Edge Framework for Sharing Economy Services in a Smart City	Blockchain and IoT-Based	Rahman, Md Abdur and Rashid, Md Mamunur and Hossain, M Shamim and Hassanain, Elham and Alhamid, Mohammed F and Guizani, Mohsen	2019
PS60	Enhancing IoT Security and Privacy Using Distributed Ledgers with IOTA and the Tangle	IoT Security and Privacy	Shabandri, Bilal and Maheshwari, Piyush	2019

Continued on next page

Table A.1 – *Continued from previous page*

ID	Title	Short name	Author(s)	Year
PS61	A User Authentication Scheme of IoT Devices using Blockchain-Enabled Fog Nodes	Authentication IoT	Almadhoun, Randa and Kadadha, Maha and Alhemeiri, Maya and Alshehhi, Maryam and Salah, Khaled	2018
PS62	Using Blockchain to Support Data and Service Management in IoV/IoT	Blockchain for data	Odiete, Obaro and Lomotey, Richard K and Deters, Ralph	2017
PS63	Blockchain and the Internet of Things: A Software Architecture Perspective	BC-IoT	Liao, Chun-Feng and Hung, Chien-Che and Chen, Kung	2019
PS64	Managing IoT devices using blockchain platform	Managing IoT	Huh, Seyoung and Cho, Sangrae and Kim, Soohyung	2017
PS65	Work-in-progress: Integrating low-power IoT devices to a Blockchain-Based Infrastructure	Work-in-progress	Özyılmaz, Kazım Rıfat and Yurdakul, Arda	2017
PS66	Edge Computing and Caching based Blockchain IoT Network	Edge and Caching	Xu, Fangmin and Yang, Fan and Zhao, Chenglin and Fang, Chao	2018

Continued on next page

Table A.1 – *Continued from previous page*

ID	Title	Short name	Author(s)	Year
PS67	Controlchain: Blockchain as a central enabler for access control authorizations in the IoT	Controlchain	Pinno, Otto Julio Ahlert and Gregio, Andre Ricardo Abed and De Bona, Luis CEe	2017
PS68	Autonomic Identity Framework for the Internet of Things	Autonomic identity	Zhu, Xiaoyang and Badr, Youakim and Pacheco, Jesus and Hariri, Salim	2017
PS69	Blockchain based credibility verification method for IoT entities	Credibilty verification verification	Qu, Chao and Tao, Ming and Zhang, Jie and Hong, Xiaoyu and Yuan, Ruifen	2018
PS70	Blockchain based data integrity service framework for IoT data	Data integrity services	Liu, Bin and Yu, Xiao Liang and Chen, Shiping and Xu, Xiwei and Zhu, Liming	2017
PS71	Blockchain Connected Gateway for BLE-Based Devices in the Internet of Things	Gateway for BLE	Cha, Shi-Cho and Chen, Jyun-Fu and Su, Chunhua and Yeh, Kuo-Hui	2018

Continued on next page

Table A.1 – *Continued from previous page*

ID	Title	Short name	Author(s)	Year
PS72	Blockchain-based dynamic key management for heterogeneous intelligent transportation systems	Key management for transportation	Lei, Ao and Cruickshank, Haitham and Cao, Yue and Asuquo, Philip and Ogah, Chibueze P Anyigor and Sun, Zhili	2018
PS73	Blockchain-based fair three-party contract signing protocol for fog computing	Three party contract for fog	Huang, Hui and Li, Kuan-Ching and Chen, Xiaofeng	2018
PS74	FairAccess: a new Blockchain-based access control framework for the Internet of Things	FairAccess	Ouaddah, Aafaf and Abou Elkalam, Anas and Ait Ouahman, Abdellah	2016
PS75	Blockchain Meets IoT: An Architecture for Scalable Access Management in IoT	Blockchain Meets IoT	Novo, Oscar	2018
PS76	Toward a blockchain cloud manufacturing system as a peer to peer distributed network platform	Cloud manufacturing	Li, Zhi and Barenji, Ali Vatankhah and Huang, George Q	2018

Continued on next page

Table A.1 – *Continued from previous page*

ID	Title	Short name	Author(s)	Year
PS77	Internet of Smart Things - IoST: Using Blockchain and CLIPS to Make Things Autonomous	IoST	Samaniego, Mayra and Deters, Ralph	2017
PS78	Using blockchain to push software-defined IoT components onto edge hosts	Software-defined IoT components	Samaniego, Mayra and Deters, Ralph	2016
PS79	Bubbles of Trust: A decentralized blockchain-based authentication system for IoT	Bubbles of Trust	Hammi, Mohamed Tahar and Hammi, Badis and Bellot, Patrick and Serhrouchni, Ahmed	2018
PS80	IoT data privacy via blockchains and IPFS	IoT data privacy	Ali, Muhammad Salek and Dolui, Koustabh and Antonelli, Fabio	2017
PS81	The IoT electric business model: Using blockchain technology for the internet of things	IoT electric business model	Zhang, Yu and Wen, Jiangtao	2017

Continued on next page

Table A.1 – *Continued from previous page*

ID	Title	Short name	Author(s)	Year
PS82	Using Ethereum Blockchain in Internet of Things: A Solution for Electric Vehicle Battery Refueling	Blockchain for Electrical Vehicles	Sun, Haoli and Hua, Song and Zhou, Ence and Pi, Bingfeng and Sun, Jun and Yamashita, Kazuhiro	2018
PS83	Using Blockchain for IOT Access Control and Authentication Management	IOT Access Control and Authentication	Ourad, Abdallah Zoubir and Belgacem, Boutheyna and Salah, Khaled	2018
PS84	Decentralized, blockchain based access control framework for the heterogeneous internet of things	Blockchain based access control	Dukkipati, Chethana and Zhang, Yunpeng and Cheng, Liang Chieh	2018
PS85	Mind my value: A decentralized infrastructure for fair and trusted IoT data trading	Mind my value	Missier, Paolo and Bajoudah, Shaimaa and Capossele, Angelo and Gaglione, Andrea and Nati, Michele	2017
PS86	Toward open manufacturing	Toward open manufacturing	Li, Zhi and Wang, WM and Liu, Guo and Liu, Layne and He, Jiadong and Huang, GQ	2018

Continued on next page

Table A.1 – *Continued from previous page*

ID	Title	Short name	Author(s)	Year
PS87	Toward a robust security paradigm for bluetooth low energy-based smart objects in the Internet-of-Things	Security paradigm for bluetooth	Cha, Shi-Cho and Yeh, Kuo-Hui and Chen, Jyun-Fu	2017
PS88	Smart contract-based access control for the internet of things	Smart contract-based access control	Zhang, Yuanyu and Kasahara, Shoji and Shen, Yulong and Jiang, Xiaohong and Wan, Jianxiong	2018
PS89	Creditcoin: A privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles	Creditcoin	Li, Lun and Liu, Jiqiang and Cheng, Lichen and Qiu, Shuo and Wang, Wei and Zhang, Xiangliang and Zhang, Zonghua	2018
PS90	Patch transporter: Incentivized, decentralized software patch system for WSN and IoT environments	Patch transporter	Lee, JongHyup	2018

Continued on next page

Table A.1 – *Continued from previous page*

ID	Title	Short name	Author(s)	Year
PS91	A sustainable home energy prosumer-chain methodology with energy tags over the blockchain	Home energy prosumer-chain	Park, Lee and Lee, Sanghoon and Chang, Hangbae	2018
PS92	A hardware-based caching system on FPGA NIC for Blockchain	A hardware-based caching	Sakakibara, Yuma and Morishima, Shin and Nakamura, Kohei and Matsutani, Hiroki	2018
PS93	Semantic blockchain to improve scalability in the internet of things	Semantic blockchain	Ruta, Michele and Scioscia, Floriano and Ieva, Saverio and Capurso, Giovanna and Di Sciascio, Eugenio	2017
PS94	Beekeeper: A blockchain-based iot system with secure storage and homomorphic computation	Beekeeper	Zhou, Lijing and Wang, Licheng and Sun, Yiru and Lv, Pin	2018
PS95	Blockchain based decentralized management of demand response programs in smart energy grids	Decentralized management of demand response	Pop, Claudia and Cioara, Tudor and Antal, Marcel and Anghel, Ionut and Salomie, Ioan and Bertoncini, Massimo	2018

Continued on next page

Table A.1 – *Continued from previous page*

ID	Title	Short name	Author(s)	Year
PS96	Smart-toy-edge-computing-oriented data exchange based on blockchain	Smart-toy-edge-computing	Yang, Jian and Lu, Zhihui and Wu, Jie	2018
PS97	A blockchain-based Trust System for the Internet of Things	A blockchain-based Trust	Di Pietro, Roberto and Salleras, Xavier and Signorini, Matteo and Waisbard, Erez	2018
PS98	Privacy-preserving and efficient aggregation based on blockchain for power grid communications in smart communities	Privacy-preserving	Guan, Zhitao and Si, Guanlin and Zhang, Xiaosong and Wu, Longfei and Guizani, Nadra and Du, Xiaojiang and Ma, Yinglong	2018
PS99	Continuous patient monitoring with a patient centric agent: A block architecture	Continuous patient monitoring	Uddin, Md Ashraf and Stranieri, Andrew and Gondal, Iqbal and Balasubramanian, Venki	2018
PS100	Blockchain-oriented coalition formation by cps resources: Ontological approach and case study	Blockchain-oriented coalition	Kashevnik, Alexey and Teslya, Nikolay	2018

Bibliography

[1] Gregory D Abowd et al. "Towards a better understanding of context and context-awareness". In: *International symposium on handheld and ubiquitous computing*. Springer. 1999, pp. 304–307.

[2] Muhammad Salek Ali et al. "Applications of blockchains in the Internet of Things: A comprehensive survey". In: *IEEE Communications Surveys & Tutorials* 21.2 (2018), pp. 1676–1717.

[3] Abdollah Ansari and Azuraliza Abu Bakar. "A comparative study of three artificial intelligence techniques: genetic algorithm, neural network, and fuzzy logic, on scheduling problem". In: *Artificial Intelligence with Applications in Engineering and Technology (ICAIET), 2014 4th International Conference on*. IEEE. 2014, pp. 31–36.

[4] Luigi Atzori, Antonio Iera, and Giacomo Morabito. "The internet of things: A survey". In: *Computer networks* 54.15 (2010), pp. 2787–2805.

[5] Marcella Atzori. "Blockchain-based architectures for the internet of things: a survey". In: (2017).

[6] Felix Bachmann, Len Bass, and Mark Klein. *Deriving architectural tactics: A step toward methodical architectural design*. Tech. rep. Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst, 2003.

[7] Arshdeep Bahga and Vijay K Madisetti. "Blockchain platform for industrial internet of things". In: *Journal of Software Engineering and Applications* 9.10 (2016), p. 533.

[8] Ying Bai and Dali Wang. "Fundamentals of fuzzy logic control—fuzzy sets, fuzzy rules and defuzzifications". In: *Advanced Fuzzy Logic Technologies in Industrial Applications*. Springer, 2006, pp. 17–36.

[9] James F Baldwin and Dong Walter Xie. "Simple fuzzy logic rules based on fuzzy decision tree for classification and prediction problem". In: *International Conference on Intelligent Information Processing*. Springer. 2004, pp. 175–184.

[10] Ahmed Banafa. "IoT standardization and implementation challenges". In: *IEEE Internet of Things Newsletter* (2016).

[11] Mario Barbacci et al. *Quality Attributes.* Tech. rep. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, 1995.

[12] Len Bass, Paul Clements, and Rick Kazman. *Software Architecture in Practice*. Third. Addison-Wesley Professional, 2012.

[13] PerOlof Bengtsson et al. "Architecture-level modifiability analysis (ALMA)". In: *Journal of Systems and Software* 69.1-2 (2004), pp. 129–147.

[14] Suman Sankar Bhunia, Sourav Kumar Dhar, and Nandini Mukherjee. "iHealth: A Fuzzy approach for provisioning Intelligent Health-care system in Smart City". In: *2014 IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE. 2014, pp. 187–193.

[15] Bitcoin. "Bitcoin Developer Guide". In: (2017).

[16] James J Buckley and Yoichi Hayashi. "Fuzzy neural networks: A survey". In: *Fuzzy sets and systems* 66.1 (1994), pp. 1–13.

[17] Harry B Burke et al. "Artificial neural networks improve the accuracy of cancer survival prediction". In: *Cancer* 79.4 (1997), pp. 857–862.

[18] Carole Cadwalladr and Emma Graham-Harrison. "Revealed: 50 million Facebook profiles harvested for Cambridge Analytica in major data breach". In: *The guardian* 17 (2018), p. 22.

[19] Donald T Campbell and Donald W Fiske. "Convergent and discriminant validation by the multitrait-multimethod matrix." In: *Psychological bulletin* 56.2 (1959), p. 81.

[20] Humberto Cervantes and Rick Kazman. *Designing Software Architectures: A Practical Approach*. Addison-Wesley, 2016.

[21] Ting Chen et al. "Dataether: Data exploration framework for ethereum". In: *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE. 2019, pp. 1369–1380.

[22] Ching-Chin Chern, Yu-Jen Chen, and Bo Hsiao. "Decision tree–based classifier in providing telehealth service". In: *BMC medical informatics and decision making* 19.1 (2019), p. 104.

[23] Konstantinos Christidis and Michael Devetsikiotis. "Blockchains and smart contracts for the internet of things". In: *Ieee Access* 4 (2016), pp. 2292–2303.

[24] Paul Clements, Rick Kazman, Mark Klein, et al. *Evaluating software architectures*. Tsinghua University Press Beijing, 2003.

[25] Marco Conoscenti, Antonio Vetro, and Juan Carlos De Martin. "Blockchain for the Internet of Things: A systematic literature review". In: *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*. IEEE. 2016, pp. 1–6.

[26] Guillermo Cueva-Fernandez et al. "Fuzzy decision method to improve the information exchange in a vehicle sensor tracking system". In: *Applied Soft Computing* 35 (2015), pp. 708–716.

[27] Hong-Ning Dai, Zibin Zheng, and Yan Zhang. "Blockchain for Internet of Things: A survey". In: *IEEE Internet of Things Journal* 6.5 (2019), pp. 8076–8094.

[28] Thanh Long Nhat Dang and Minh Son Nguyen. "An approach to data privacy in smart home using blockchain technology". In: *2018 International Conference on Advanced Computing and Applications (ACOMP)*. IEEE. 2018, pp. 58–64.

[29] Thomas J Dolan. "Architecture assessment of information-system families". In: *Eindhoven University of Technology Ph. D Thesis, Department of Technology Management, Eindhoven* (2002).

[30] Ali Dorri, Salil S Kanhere, and Raja Jurdak. "Towards an optimized blockchain for IoT". In: *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*. ACM. 2017, pp. 173–178.

[31] Ali Dorri et al. "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)*. IEEE. 2017, pp. 618–623.

[32] Alevtina Dubovitskaya et al. "Secure and trustable electronic medical records sharing using blockchain". In: *AMIA Annual Symposium Proceedings*. Vol. 2017. American Medical Informatics Association. 2017, p. 650.

[33] Joaquin Durán et al. "Obstructive sleep apnea–hypopnea and related clinical features in a population-based sample of subjects aged 30 to 70 yr". In: *American journal of respiratory and critical care medicine* 163.3 (2001), pp. 685–689.

[34] Jacob Eberhardt and Stefan Tai. "On or off the blockchain? Insights on off-chaining computation and data". In: *European Conference on Service-Oriented and Cloud Computing*. Springer. 2017, pp. 3–15.

[35] Christian Esposito et al. "Blockchain: A panacea for healthcare cloud-based data security and privacy?" In: *IEEE Cloud Computing* 5.1 (2018), pp. 31–37.

[36] Chin-Yuan Fan et al. “A hybrid model combining case-based reasoning and fuzzy decision tree for medical data classification”. In: *Applied Soft Computing* 11.1 (2011), pp. 632–644.

[37] Tiago M Fernández-Caramés and Paula Fraga-Lamas. “A Review on the Use of Blockchain for the Internet of Things”. In: *IEEE Access* 6 (2018), pp. 32979–33001.

[38] Mohamed Amine Ferrag et al. “Blockchain technologies for the internet of things: Research issues and challenges”. In: *IEEE Internet of Things Journal* 6.2 (2018), pp. 2188–2204.

[39] Eduardo Flores-Morán, Wendy Yánez-Pazmiño, and Julio Barzola-Monteses. “Genetic algorithm and fuzzy self-tuning PID for DC motor position controllers”. In: *2018 19th International Carpathian Control Conference (ICCC)*. IEEE. 2018, pp. 162–168.

[40] Ala Al-Fuqaha et al. “Internet of things: A survey on enabling technologies, protocols, and applications”. In: *IEEE communications surveys & tutorials* 17.4 (2015), pp. 2347–2376.

[41] David Garlan and Mary Shaw. “An introduction to software architecture”. In: *Advances in software engineering and knowledge engineering*. World Scientific, 1993, pp. 1–39.

[42] David Garland and Mary Shaw. “An introduction to software architecture”. In: *Advances in Software Engineering and Knowledge Engineering* 1 (1993).

[43] Jeff Garzik. “Public versus private blockchains”. In: *BitFury Group, San Francisco, USA, White Paper* 1 (2015).

[44] Larysa Globa et al. “Fuzzy logic usage for the data processing in the Internet of Things networks”. In: (2018).

[45] Kristen N Griggs et al. "Healthcare blockchain system using smart contracts for secure automated remote patient monitoring". In: *Journal of medical systems* 42.7 (2018), p. 130.

[46] Güney Gürsel et al. "Healthcare, uncertainty, and fuzzy logic". In: *Digital Medicine* 2.3 (2016), p. 101.

[47] Kashif Hameed et al. "An intelligent IoT based healthcare system using fuzzy neural networks". In: *Scientific Programming* 2020 (2020).

[48] Mohamed Tahar Hammi et al. "Bubbles of Trust: A decentralized blockchain-based authentication system for IoT". In: *Computers & Security* 78 (2018), pp. 126–142.

[49] Sarra Hammoudi, Zibouda Aliouat, and Saad Harous. "Challenges and research directions for Internet of Things". In: *Telecommunication Systems* 67.2 (2018), pp. 367–385.

[50] Neil B Harrison and Paris Avgeriou. "How do architecture patterns and tactics interact? A model and annotation". In: *Journal of Systems and Software* 83.10 (2010), pp. 1735–1758.

[51] Anne-Wil Harzing. *The publish or perish book*. Tarma Software Research Pty Limited, 2010.

[52] Jordi Herrera-Joancomartı and Cristina Pérez-Solà. "Privacy in bitcoin transactions: new challenges from blockchain scalability solutions". In: *International Conference on Modeling Decisions for Artificial Intelligence*. Springer. 2016, pp. 26–44.

[53] Seyoung Huh, Sangrae Cho, and Soohyung Kim. "Managing IoT devices using blockchain platform". In: *2017 19th international conference on advanced communication technology (ICACT)*. IEEE. 2017, pp. 464–467.

[54] M Usman Iftikhar et al. "DeltaIoT: A self-adaptive Internet of Things exemplar". In: *Proceedings of the 12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*. IEEE Press. 2017, pp. 76–82.

[55] Didac Gil De La Iglesia and Danny Weyns. "MAPE-K formal templates to rigorously design behaviors for self-adaptive systems". In: *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 10.3 (2015), pp. 1–31.

[56] Su-Hwan Jang et al. "Fog Computing Architecture Based Blockchain for Industrial IoT". In: *International Conference on Computational Science*. Springer. 2019, pp. 593–606.

[57] Yiming Jiang et al. "A cross-chain solution to integrating multiple blockchains for IoT data management". In: *Sensors* 19.9 (2019), p. 2042.

[58] Hai Jin, Xiaohai Dai, and Jiang Xiao. "Towards a novel architecture for enabling interoperability amongst multiple blockchains". In: *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE. 2018, pp. 1203–1211.

[59] *Joulemeter: Computational Energy Measurement and Optimization*. https://www.microsoft.com/en-us/research/project/joulemetercomputational-energy-measurement-and-optimization. Accessed: 2019-11-20.

[60] Luo Kan et al. "A multiple blockchains architecture on inter-blockchain communication". In: *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*. IEEE. 2018, pp. 139–145.

[61] Oğuz Karan et al. "Diagnosing diabetes using neural networks on small mobile devices". In: *Expert Systems with Applications* 39.1 (2012), pp. 54–60.

[62] Jaideep Kaur and Kamaljit Kaur. "A fuzzy approach for an IoT-based automated employee performance appraisal". In: *Computers, Materials and Continua* 53.1 (2017), pp. 24–38.

[63] Rick Kazman et al. “Experience with performing architecture tradeoff analysis”. In: *Proceedings of the 1999 International Conference on Software Engineering (IEEE Cat. No. 99CB37002)*. IEEE. 1999, pp. 54–63.

[64] Rick Kazman et al. “SAAM: A method for analyzing the properties of software architectures”. In: *Proceedings of 16th International Conference on Software Engineering*. IEEE. 1994, pp. 81–90.

[65] Rick Kazman et al. “The architecture tradeoff analysis method”. In: *Proceedings. Fourth IEEE International Conference on Engineering of Complex Computer Systems (Cat. No. 98EX193)*. IEEE. 1998, pp. 68–78.

[66] Manju Khari et al. “Internet of Things: Proposed security aspects for digitizing the world”. In: *2016 3rd international conference on computing for sustainable global development (INDIACom)*. IEEE. 2016, pp. 2165–2170.

[67] Deepak Khazanchi. “A Philosophical Framework for the Validation of Information Systems Concepts”. In: (1996).

[68] Zach Kirsch and Ming Chow. “Quantum Computing: The Risk to Existing Encryption Methods”. In: *Retrieved from URL: http://www. cs. tufts. edu/comp/116/archive/fall2015/zkirsch. pdf* (2015).

[69] Barbara Kitchenham and Pearl Brereton. “A systematic review of systematic review process research in software engineering”. In: *Information and software technology* 55.12 (2013), pp. 2049–2075.

[70] Barbara Kitchenham and Stuart Charters. “Guidelines for performing systematic literature reviews in software engineering”. In: (2007).

[71] Nir Kshetri. “Can blockchain strengthen the internet of things?” In: *IT professional* 19.4 (2017), pp. 68–72.

[72] Boohyung Lee and Jong-Hyouk Lee. "Blockchain-based secure firmware update for embedded devices in an Internet of Things environment". In: *The Journal of Supercomputing* 73.3 (2017), pp. 1152–1167.

[73] Grace Lewis and Patricia Lago. "Architectural tactics for cyber-foraging: Results of a systematic literature review". In: *Journal of Systems and Software* 107 (2015), pp. 158–186.

[74] Jianan Li et al. "Decentralized On-Demand Energy Supply for Blockchain in Internet of Things: A Microgrids Approach". In: *IEEE Transactions on Computational Social Systems* (2019).

[75] Zhetao Li et al. "Consortium blockchain for secure energy trading in industrial internet of things". In: *IEEE transactions on industrial informatics* 14.8 (2017), pp. 3690–3700.

[76] Xueping Liang et al. "Integrating blockchain for data sharing and collaboration in mobile healthcare applications". In: *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE. 2017, pp. 1–5.

[77] Xueping Liang et al. "Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability". In: *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. IEEE Press. 2017, pp. 468–477.

[78] Chun-Feng Liao, Chien-Che Hung, and Kung Chen. "Blockchain and the Internet of Things: A Software Architecture Perspective". In: *Business Transformation through Blockchain*. Springer, 2019, pp. 53–75.

[79] Chun-Feng Liao et al. "On design issues and architectural styles for blockchain-driven IoT services". In: *2017 IEEE international conference on consumer electronics-Taiwan (ICCE-TW)*. IEEE. 2017, pp. 351–352.

[80] Tri Listyorini and Robbi Rahim. "A prototype fire detection implemented using the Internet of Things and fuzzy logic". In: *World Trans. Eng. Technol. Educ* 16.1 (2018), pp. 42–46.

[81] Sin Kuang Lo et al. "Analysis of Blockchain Solutions for IoT: A Systematic Literature Review". In: *IEEE Access* 7 (2019), pp. 58822–58835. DOI: 10.1109/ACCESS.2019.2914675. URL: https://doi.org/10.1109/ACCESS.2019.2914675.

[82] Joan Albert López-Vallverdú, David RiañO, and John A Bohada. "Improving medical decision trees by combining relevant health-care criteria". In: *Expert Systems with Applications* 39.14 (2012), pp. 11782–11791.

[83] David J Lowe and Jamshid Parvar. "A logistic regression approach to modelling the contractor's decision to bid". In: *Construction Management and Economics* 22.6 (2004), pp. 643–653.

[84] Qinghua Lu and Xiwei Xu. "Adaptable blockchain-based systems: A case study for product traceability". In: *IEEE Software* 34.6 (2017), pp. 21–27.

[85] Loi Luu et al. "A secure sharding protocol for open blockchains". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM. 2016, pp. 17–30.

[86] Andrzej Magruk. "The most important aspects of uncertainty in the Internet of Things field–context of smart buildings". In: *Procedia Engineering* 122 (2015), pp. 220–227.

[87] Parikshit N Mahalle et al. "A fuzzy approach to trust based access control in internet of things". In: *Wireless VITAE 2013*. IEEE. 2013, pp. 1–5.

[88] Redowan Mahmud et al. "Quality of Experience (QoE)-aware placement of applications in Fog computing environments". In: *Journal of Parallel and Distributed Computing* (2018).

[89] Imran Makhdoom, Mehran Abolhasan, and Wei Ni. "Blockchain for IoT: The Challenges and aWay Forward". In: *Proceedings of the 15th International Joint Conference on e-Business and Telecommunications-Volume 2: SECRYPT*. INSTICC. 2018.

[90] Ebrahim H Mamdani and Sedrak Assilian. "An experiment in linguistic synthesis with a fuzzy logic controller". In: *International journal of man-machine studies* 7.1 (1975), pp. 1–13.

[91] Daniel Meana-Llorián et al. "IoFClime: The fuzzy logic and the Internet of Things to control indoor temperature regarding the outdoor ambient conditions". In: *Future Generation Computer Systems* 76 (2017), pp. 275–284.

[92] Renzo Modica and Angelo Rizzo. "Accuracy and response time of a portable pulse oximeter". In: *Respiration* 58.3-4 (1991), pp. 155–157.

[93] Manuel Eduardo Flores Morán and Nataly Aracely Pozo Viera. "Comparative study for DC motor position controllers". In: *Ecuador Technical Chapters Meeting (ETCM), 2017 IEEE*. IEEE. 2017, pp. 1–6.

[94] Henry Muccini and Mahyar Tourchi Moghaddam. "Iot architectural styles". In: *European Conference on Software Architecture*. Springer. 2018, pp. 68–85.

[95] Jianbing Ni et al. "Securing fog computing for internet of things applications: Challenges and solutions". In: *IEEE Communications Surveys & Tutorials* 20.1 (2017), pp. 601–628.

[96] Ashner Gerald P Novilla, August Anthony N Balute, and Dennis B Gonzales. "The Use of Fuzzy Logic for Online Monitoring of Manufacturing Machine: An Intelligent System". In: (2017).

[97] Lucila Ohno-Machado, Ronilda Lacson, and Eduardo Massad. "Decision trees and fuzzy logic: a comparison of models for the selection of measles vaccination strategies

in Brazil." In: *Proceedings of the AMIA Symposium*. American Medical Informatics Association. 2000, p. 625.

[98] Alfonso Panarello et al. "Blockchain and iot integration: A systematic survey". In: *Sensors* 18.8 (2018), p. 2575.

[99] Arpit Patel and Tushar A Champaneria. "Fuzzy logic based algorithm for Context Awareness in IoT for Smart home environment". In: *2016 IEEE Region 10 Conference (TENCON)*. IEEE. 2016, pp. 1057–1060.

[100] Clements Paul, Rick Kazman, and Mark Klein. "Evaluating software architectures: methods and case studies". In: *AddisonÇęWesley, Boston, MA, USA* (2002).

[101] Ken Peffers et al. "A design science research methodology for information systems research". In: *Journal of management information systems* 24.3 (2007), pp. 45–77.

[102] Kai Petersen, Sairam Vakkalanka, and Ludwik Kuzniarz. "Guidelines for conducting systematic mapping studies in software engineering: An update". In: *Information and Software Technology* 64 (2015), pp. 1–18.

[103] Stacie Petter, Deepak Khazanchi, and John D Murphy. "A design science based evaluation framework for patterns". In: *ACM SIGMIS Database: the DATABASE for Advances in Information Systems* 41.3 (2010), pp. 9–26.

[104] Joseph Poon and Vitalik Buterin. "Plasma: Scalable autonomous smart contracts". In: *White paper* (2017), pp. 1–47.

[105] Addison-Wesley Professional, ed. *Arquitectura de software en la práctica*.

[106] Deepak Puthal et al. "Threats to networking cloud and edge datacenters in the Internet of Things". In: *IEEE Cloud Computing* 3.3 (2016), pp. 64–71.

[107] Han Qiu et al. "A Dynamic Scalable Blockchain Based Communication Architecture for IoT". In: *International Conference on Smart Blockchain*. Springer. 2018, pp. 159–166.

[108] Tie Qiu et al. "How can heterogeneous internet of things build our future: A survey". In: *IEEE Communications Surveys & Tutorials* 20.3 (2018), pp. 2011–2027.

[109] Md Abdur Rahman et al. "Blockchain and IoT-based cognitive edge framework for sharing economy services in a smart city". In: *IEEE Access* 7 (2019), pp. 18611–18621.

[110] Mohammad Abdur Razzaque et al. "Middleware for Internet of Things: A survey." In: *IEEE Internet of Things Journal* 3.1 (2016), pp. 70–95.

[111] Ana Reyna et al. "On blockchain and its integration with IoT. Challenges and opportunities". In: *Future Generation Computer Systems* 88 (2018), pp. 173–190.

[112] John R Rossiter. "The C-OAR-SE procedure for scale development in marketing". In: *International journal of research in marketing* 19.4 (2002), pp. 305–335.

[113] Gokhan Sagirlar et al. "Hybrid-iot: Hybrid blockchain architecture for internet of things-pow sub-blockchains". In: *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE. 2018, pp. 1007–1016.

[114] Mayra Samaniego and Ralph Deters. "Blockchain as a Service for IoT". In: *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE. 2016, pp. 433–436.

[115] Mayra Samaniego and Ralph Deters. "Using blockchain to push software-defined IoT components onto edge hosts". In: *Proceedings of the International Conference on Big Data and Advanced Wireless Technologies*. ACM. 2016, p. 58.

[116] Amilcare Francesco Santamaria et al. "A two stages fuzzy logic approach for Internet of Things (IoT) wearable devices". In: *2016 IEEE 27th Annual International Sympo-*

sium on Personal, Indoor, and Mobile Radio Communications (PIMRC). IEEE. 2016, pp. 1–6.

[117] David Sarabia-Jácome et al. "Energy consumption in software defined networks to provide service for mobile users". In: *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE. 2017, pp. 1453–1458.

[118] André Schweizer et al. "Unchaining Social Businesses-Blockchain as the Basic Technology of a Crowdlending Platform." In: *ICIS*. 2017.

[119] James Scott and Rick Kazman. *Realizing and refining architectural tactics: Availability*. Tech. rep. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, 2009.

[120] Bilal Shabandri and Piyush Maheshwari. "Enhancing IoT Security and Privacy Using Distributed Ledgers with IOTA and the Tangle". In: *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*. IEEE. 2019, pp. 1069–1075.

[121] Hossein Shafagh et al. "Towards blockchain-based auditable storage and sharing of IoT data". In: *Proceedings of the 2017 on Cloud Computing Security Workshop*. ACM. 2017, pp. 45–50.

[122] Pradip Kumar Sharma, Mu-Yen Chen, and Jong Hyuk Park. "A software defined fog node based distributed blockchain cloud architecture for IoT". In: *IEEE Access* 6 (2017), pp. 115–124.

[123] Pradip Kumar Sharma and Jong Hyuk Park. "Blockchain based hybrid network architecture for the smart city". In: *Future Generation Computer Systems* 86 (2018), pp. 650–655.

[124] Surendra Kumar Sharma et al. "Prevalence and risk factors of obstructive sleep apnea syndrome in a population of Delhi, India". In: *Chest* 130.1 (2006), pp. 149–156.

[125] PG Shynu et al. "Blockchain-based secure healthcare application for diabetic-cardio disease prediction in fog computing". In: *IEEE Access* 9 (2021), pp. 45706–45720.

[126] Ian Sommerville. *Engineering software products*. Pearson, 2020.

[127] Natalie L Sproull. *Handbook of research methods: A guide for practitioners and students in the social sciences*. Scarecrow press, 2002.

[128] Alexandru Stanciu. "Blockchain based distributed control system for edge computing". In: *2017 21st International Conference on Control Systems and Computer Science (CSCS)*. IEEE. 2017, pp. 667–671.

[129] Detmar Straub, Marie-Claude Boudreau, and David Gefen. "Validation guidelines for IS positivist research". In: *Communications of the Association for Information systems* 13.1 (2004), p. 24.

[130] Detmar W Straub. "Validating instruments in MIS research". In: *MIS quarterly* (1989), pp. 147–169.

[131] Safieh Tahmasebipour and Seyed Morteza Babamir. "Ranking of common architectural styles based on availability, security and performance quality attributes". In: (2014).

[132] Nikolay Teslya and Igor Ryabchikov. "Blockchain-based platform architecture for industrial IoT". In: *2017 21st Conference of Open Innovations Association (FRUCT)*. IEEE. 2017, pp. 321–329.

[133] Feng Tian. "An agri-food supply chain traceability system for China based on RFID & blockchain technology". In: *Service Systems and Service Management (ICSSSM), 2016 13th International Conference on*. IEEE. 2016, pp. 1–6.

[134] OĞUZ HAN TİMUŞ and EMİNE DOĞRU BOLAT. "k-NN-based classification of sleep apnea types using ECG". In: *Turkish Journal of Electrical Engineering & Computer Sciences* 25.4 (2017), pp. 3008–3023.

[135] Hua Ting et al. "Decision tree based diagnostic system for moderate to severe obstructive sleep apnea". In: *Journal of medical systems* 38.9 (2014), p. 94.

[136] William MK Trochim and James P Donnelly. *Research methods knowledge base*. Vol. 2. Atomic Dog Pub., 2001.

[137] Shreshth Tuli et al. "FogBus: A Blockchain-based Lightweight Framework for Edge and Fog Computing". In: *arXiv preprint arXiv:1811.11978* (2018).

[138] Shreshth Tuli et al. "Fogbus: A blockchain-based lightweight framework for edge and fog computing". In: *Journal of Systems and Software* 154 (2019), pp. 22–36.

[139] KS Vani and Rajesh Rayappa Neeralagi. "IoT based health monitoring using fuzzy logic". In: *Int. J. Comput. Intell. Res* 13.10 (2017), pp. 2419–2429.

[140] Shuai Wang et al. "Blockchain-enabled smart contracts: architecture, applications, and future trends". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 49.11 (2019), pp. 2266–2277.

[141] Danny Weyns, Gowri Sankar Ramachandran, and Ritesh Kumar Singh. "Self-managing internet of things". In: *International Conference on Current Trends in Theory and Practice of Informatics*. Springer. 2018, pp. 67–84.

[142] Claes Wohlin. "Guidelines for snowballing in systematic literature studies and a replication in software engineering". In: *Proceedings of the 18th international conference on evaluation and assessment in software engineering*. Citeseer. 2014, p. 38.

[143] Dominic Wörner and Thomas von Bomhard. "When your sensor earns money: exchanging data for cash with Bitcoin". In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. ACM. 2014, pp. 295–298.

[144] Longfei Wu et al. "An out-of-band authentication scheme for internet of things using blockchain technology". In: *2018 International Conference on Computing, Networking and Communications (ICNC)*. IEEE. 2018, pp. 769–773.

[145] Mingli Wu et al. "A Comprehensive Survey of Blockchain: From Theory to IoT Applications and Beyond". In: *IEEE Internet of Things Journal* 6.5 (2019), pp. 8114–8154. DOI: 10.1109/JIOT.2019.2922538. URL: https://doi.org/10.1109/JIOT.2019.2922538.

[146] QI Xia et al. "MeDShare: Trust-less medical data sharing among cloud service providers via blockchain". In: *IEEE Access* 5 (2017), pp. 14757–14767.

[147] Zehui Xiong et al. "The best of both worlds: A general architecture for data management in blockchain-enabled Internet-of-Things". In: *IEEE Network* 34.1 (2020), pp. 166–173.

[148] Fangmin Xu et al. "Edge Computing and Caching based Blockchain IoT Network". In: *2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)*. IEEE. 2018, pp. 238–239.

[149] Xiwei Xu et al. "A pattern collection for blockchain-based applications". In: *Proceedings of the 23rd European Conference on Pattern Languages of Programs*. ACM. 2018, p. 3.

[150] Xiwei Xu et al. "A taxonomy of blockchain-based systems for architecture design". In: *Software Architecture (ICSA), 2017 IEEE International Conference on*. IEEE. 2017, pp. 243–252.

[151] Xiwei Xu et al. "The blockchain as a software connector". In: *Software Architecture (WICSA), 2016 13th Working IEEE/IFIP Conference on*. IEEE. 2016, pp. 182–191.

[152] Wendy Yánez et al. "Architecting Internet of Things Systems with Blockchain: A Catalog of Tactics". In: *ACM Transactions on Software Engineering and Methodology (TOSEM)* 30.3 (2021), pp. 1–46.

[153] Wendy Yánez et al. “Data allocation mechanism for Internet-of-Things systems with blockchain”. In: *IEEE Internet of Things Journal* 7.4 (2020), pp. 3509–3522.

[154] Wendy Yánez et al. “Reference Arquitectural styles for IoT systems supported by Blockchain”. unpublished. 2022.

[155] Kimchai Yeow et al. “Decentralized consensus for edge-centric internet of things: A review, taxonomy, and research issues”. In: *IEEE Access* 6 (2017), pp. 1513–1524.

[156] Bessie Ann Young et al. “Diabetes complications severity index and risk of mortality, hospitalization, and healthcare utilization”. In: *The American journal of managed care* 14.1 (2008), p. 15.

[157] Yong Yuan and Fei-Yue Wang. “Towards blockchain-based intelligent transportation systems”. In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2016, pp. 2663–2668.

[158] Zibin Zheng et al. “An overview of blockchain technology: Architecture, consensus, and future trends”. In: *Big Data (BigData Congress), 2017 IEEE International Congress on*. IEEE. 2017, pp. 557–564.

[159] Michael Herbert Ziegler, Marcel Groβmann, and Udo R Krieger. “Integration of Fog Computing and Blockchain Technology Using the Plasma Framework”. In: *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE. 2019, pp. 120–123.