A Literature Study on Application Domains and IoT Software Systems Architectures Solutions Influencing Quality Requirements

Fernando N. R. da Silva, Bruno P. de Souza, Guilherme H. Travassos

PESC/COPPE - Federal University of Rio de Janeiro (UFRJ)
Rio de Janeiro – RJ – Brazil

{fernandonrs,bpsouza,ght}@cos.ufrj.br

Abstract. *The Internet of Things (IoT) enables the development of software systems using exclusively addressable objects. This literature study investigates IoT software systems' architectural models and quality requirements. The study reveals* ***28*** *architectural solutions in* ***four*** *application domains, influencing* ***seven*** *quality requirements and indicating best practices that can be used to support decision-making when engineering IoT software systems.*

Resumo. *A Internet das Coisas (IoT) permite o desenvolvimento de sistemas de software utilizando exclusivamente objetos endereçáveis. Este estudo da literatura investiga modelos arquiteturais e requisitos de qualidade necessários para sistemas de software IoT. O estudo revelou* ***28*** *diferentes soluções de arquitetura em* ***quatro*** *domínios de aplicação, influenciando* ***sete*** *requisitos de qualidade que podem apoiar a tomada de decisão sobre melhores práticas para a engenharia de sistemas de software IoT.*

1. Introduction

Throughout history, technological advancements such as steam power, electricity, and electronics have significantly transformed our daily lives and workplaces. We are in the midst of the fourth industrial revolution, which is characterized by integrating physical objects, sensors, actuators, and the Internet. This revolution is made possible by breakthroughs such as the Internet of Things (IoT) and software solutions.

IoT was introduced by Kevin Ashton in 1999 as the interconnection of sensing and actuating devices that share information through a unified network, as mentioned in [Gubbi et al. 2013]. Atzori et al. (2010) define IoT as the widespread presence of various objects that can interact and cooperate through unique addressing schemes. Motta et al. (2019) describe IoT as a paradigm that allows the composition of software systems from exclusively addressable objects that can communicate and cooperate to achieve a goal.

Atzori et al. (2010) have identified five primary application domains for IoT: transportation, healthcare, smart environments, personal/social, and futuristic. Gubbi et al. (2013) categorized IoT application domains as personal/home, enterprise, utilities, and mobile. These domains are significant in realizing IoT software projects.

The IoT paradigm capabilities offer opportunities for creating numerous software systems that can significantly enhance our quality of life in domains like homes, travel, work, healthcare, and sports activities. Architecture is a crucial asset that facilitates the

development of intricate and high-quality software systems and mitigates potential design and implementation challenges. These interconnected systems operate intelligently based on the received information and require simplified architecture to handle diverse hardware and network protocols, ensuring smooth operation and high quality.

The quality of a software system is determined by the degree to which it meets its stakeholders' explicit and implicit requirements, thus providing value in terms of functionality, performance, security, and maintainability [ISO/IEC 25010 2023]. Architectural decisions for software systems aim to fulfill stakeholders' expectations through quality requirements (QRs) or attributes. However, these requirements are often neglected, leading to project failures, especially in IoT software systems.

However*, how do architecture IoT software systems take one or another QR into account?* Therefore, we decided to review the literature to comprehensively understand IoT software systems, exploring how various application domains and their distinct characteristics influence QRs. We want to understand architectural solutions and their impact on healthcare software projects. Our interest is driven by the need to identify the best software system arrangements. Our research question focuses on supporting software engineers in making these decisions. By analyzing the software architecture of different IoT software systems, we can gain insight into the factors contributing to their success. Therefore, it is vital to clearly understand the application domains and their corresponding software architectures to ensure the delivery of high-quality IoT software systems. Consequently, we conducted a literature study on architectural design solutions influencing QRs in IoT software systems to answer the following question:

"What application domains and characteristics of their IoT software systems architectures influence QRs?"

Our review of 28 primary studies provided a taxonomy of the most recent IoT software architectures and identified application domains, architectural solutions, and QRs. This paper offers insights and practices for engineering IoT software systems and raises research opportunities.

Besides this introduction, this paper comprises five more sections. In Section 2, related works are presented. Section 3 describes how our literature study was conducted. In section 4, the results of this study are shown. Next, section 5 shows the threats to validity. Finally, in section 6, the conclusions and future research are discussed.

2. Related Works

The ongoing study underscores the importance of continuous updates in guiding decisions for IoT software systems' architecture design. The significance of these two related works will significantly influence the research strategy. Firstly, Alreshidi and Ahmad (2019) focus on the architectural challenges in designing software systems for the Internet of Things (IoT). The authors highlight the difficulty in creating systems that can handle IoT systems' heterogeneity, dynamicity, and scalability. They also discuss the need for addressing security and privacy concerns in designing IoT systems. The article proposes a conceptual model for developing IoT software systems that integrate cloud services and open standards to overcome these architectural challenges. Also later, Razzaq (2020), presents a systematic review of software architectures for IoT systems focusing on

adopting microservices architecture. The authors emphasize the need for IoT software systems to be scalable, flexible, and handle data diversity. They review traditional architectural patterns such as n-tier and service-oriented architectures and discuss their limitations in the context of IoT. The article then explores the potential benefits of microservices architecture in addressing the architectural challenges of IoT systems, such as achieving modularization, flexibility, and compatibility. Finally, the article proposes future research directions for adopting microservices architecture in IoT systems.

Besides, it is worth discussing IoT reference architectures. A reference architecture provides a flexible framework for developing and evaluating systems, especially in the industrial IoT. It is not tied to specific technologies or standards, offering adaptable structural guidance for networks, cloud services, and compatible hardware. Experts from various fields propose these architectures to transform industries using available technologies [Mirani et al. 2022]. Specifically, reference architectures offer a basic IoT software system design, showing the main elements, hardware, and software components and how they connect to ensure the functioning and operations of an IoT-based software system [Alreshidi and Ahmad 2019]. Reference architectures are usually created informally. However, establishing a systematic design approach guarantees quality, durability, and sustainability. At the same time, many reference architectures do not survive after their first release or publication in journals or scientific events [Nakagawa and Antonio, 2023].

2.1. Industrial Internet Reference Architecture (IIRA)

The Industrial Reference Architecture for the Industrial Internet of Things (IIoT), known as the IIRA, is a model that system architects can use to design IoT software systems. It is not a standard, but it is based on the Industrial Internet Architecture Framework (IIAF), which provides principles and guidelines for the consistent description of architectures in the IIoT context. The IIRA aims to promote a shared basis of understanding and improved interoperability across industrial sectors.

The IIRA adopts a generic, high-level abstraction approach for broad applicability in industry. It consists of three layers: the Edge Layer, the Platform Layer, and the Enterprise Layer. These layers are interconnected to create a Proximity Network that combines various devices, control systems, sensors, and assets. The IIRA aims to guide the development of standards and technologies needed for interoperability and applicability in different industrial contexts, thus providing a conceptual roadmap for designing and implementing IoT software systems [Mirani et al. 2022] [Nakagawa and Antonio 2023].

2.2 Reference Architectural Model (RAMI 4.0)

RAMI 4.0 is a Reference Architectural Model for Industry 4.0, developed in Germany to modernize manufacturing and industrial automation processes. RAMI 4.0 guidelines promote an interconnected network of products, distribution of functionalities throughout the network structure, and independent communication between participants, regardless of the system hierarchy. It is based on Service Oriented Architecture (SOA) to provide networking protocols for the system components, enabling the transformation of complex tasks into simpler processes based on independent technologies and products, with

interdisciplinary collaborations among electronics, electrical, mechanical, and IT standards [Alreshidi and Ahmad 2019] [Mirani et al. 2022] [Nakagawa and Antonio 2023].

3. Literature Study

This study explores the areas where the features of IoT software system architectures emerge. The goal is to gather these projects' commonly observed QRs and comprehensively analyze the identified ones in the primary sources. These QRs encapsulate architectural challenges relevant to our IoT software projects, characterizing these IoT software systems regarding application domains and other QRs (such as security, performance, maintainability, and compatibility). This characterization is done from the perspective of software engineering researchers, drawing upon existing knowledge in technical literature related to architectural characteristics within the context of IoT software projects. Therefore, we have divided the primary research question (Section 1) into five parts to facilitate a deeper understanding (Table 1).

The research began with a Literature Search (LS) [Kuhrmann et al. 2017] to identify articles from 2019, due to Alreshidi and Ahmad (2019), in the technical literature addressing architectural design solutions influencing QRs in IoT software systems. An initial PICO-inspired search string (Table 1) was performed on Scopus (www.scopus.com) on December 22, 2022, returning 130 articles, with six selected to form the starting points for Snowballing (one level backward and forward) [Wöhlin 2014]. The result was an initial set of papers, with two being related and aligned with the objectives of our study. Subsequent searches and refinements, incorporating search terms (Table 1) from the related works [Alreshidi and Ahmad 2019] and [Razzaq 2020], led to a broad final search in Scopus in July 2023, producing 38 articles. Four articles from this set were appended to the initial set of papers, totaling ten, for executing a new Snowballing trial (one level backward and forward).

Table 1. Research Questions, Search Strings, Inclusion, and Exclusion Criteria.

RQs		**RQ1:** What are the application domains of IoT software systems? **RQ2**: What are the proposed IoT software system architectures? **RQ2.1:** What are the characteristics of these IoT software system architectures? **RQ2.2:** What are the QRs identified in these IoT software system architectures? **RQ2.3:** How are these QRs worked out in these IoT software system architectures?
Search String	**Initial**	("IoT" OR "Internet of Things") AND ("Quality Requirement" OR "Non-Functional" OR "architectural requirements") AND ("architecture" OR "architectural elements")
	Final	(software OR "Software Architect*") AND ("IoT" OR "Internet of Things") AND ("Quality Requirement" OR "Non-Functional" OR "Architectural Requirements") AND ("Architecture" OR "Architectural Elements" OR component OR design OR model OR framework) AND PUBYEAR > 2019 AND PUBYEAR < 2023
Criteria	**Inclusion**	The paper must be in the context of IoT software systems.
		The paper must report a primary study.
		The paper must provide data to answer most of the LS research questions.
		The paper must be written in the English language.
		The paper's publishing date must be from 2020 to 2022.

	Exclusion	Duplicate publication/self-plagiarism.
		Register of proceedings.
		Papers that are not peer-reviewed.

Finally, the papers were chosen based on specific inclusion and exclusion criteria (Table 1). Subsequently, two researchers thoroughly examined the chosen sources to address each research question. A final researcher evaluated the selection process, in which 28 primary sources were identified and selected.

4. Results

At the end of the study, we anticipate identifying a set of IoT software systems domains and their primary architectures, describing them, and associating them with at least one QR [ISO/IEC-25010 2023], such as security, performance, and flexibility. Thus, we aim to provide a dataset for decision-making regarding architecture solutions when designing and developing IoT software systems focusing on QRs. A mapping categorizes the selected IoT application domains, QRs, and software systems architectures. Next, these solutions are described, focusing on the addressed QRs. Finally, the architectural features for each QR are organized into catalogs.

4.1. What are the application domains of IoT software systems?

The primary sources revealed similar IoT application domains with various terminologies. Therefore, we propose a classification of IoT application domains based on [Atzori et al. 2010], [Gubbi et al. 2013], and [Motta et al. 2019].

Our study identified 28 architectural solutions associated with IoT application domains or for generic use. Some of these solutions are versatile and can be applied across all domains called "Generic." It does not mean the others cannot be used in domains other than those proposed in the primary source. Subsequently, Figure 1 presents the statistical analysis of the application areas for these architectures.

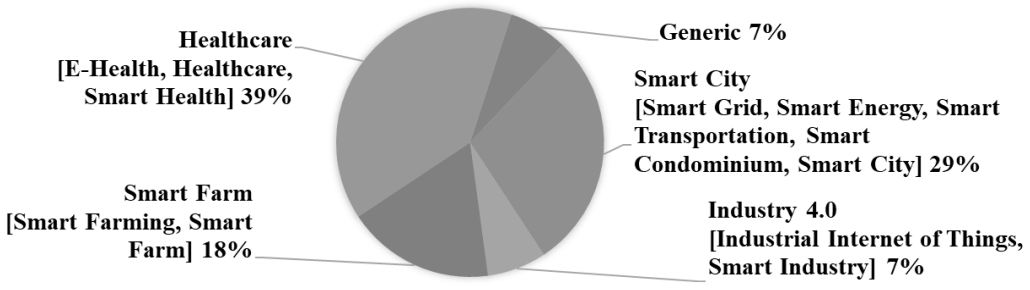

Figure 1. IoT Application Domains found in this study.

4.2. What are the proposed IoT software systems architectures?

Table 2 lists proposed IoT software systems architectures for applications such as healthcare, industry 4.0, smart cities, and smart farms. The architectures vary from cloud-based solutions to event-driven IoT, blockchain-based, geographic-based, layered, edge-computing fault-tolerant frameworks. These architectures are designed to address specific challenges related to the domain and the QR addressed. The variations reflect the

complexity of IoT software systems and the need for customized solutions to ensure efficient and successful implementation.

Table 2. IoT Software Systems Architectures Solutions.

IoT Domain	Architecture Solution	Source (Appendix A)
Generic	SDN based	[SS4]
	Layered Blockchain-Based SDN	[SS27]
Healthcare	Layered	[SS2] [SS18]
	Cloud-based Secure Biometric	[SS19]
	Event-driven IoT	[SS16]
	Deep Learning based	[SS23]
	Fog Computing based	[SS3]
	Hybrid (Centralized and Distributed)	[SS5]
	Distributed Edge-Cloud	[SS25]
	Multi-layered Cloud-Edge	[SS20]
	Hybrid Cloud-Fog Computing	[SS10]
Industry 4.0	Industrial Internet of Things (IIoT)	[SS24]
	Layered	[SS22]
Smart City	Layered Blockchain and AI-enabled	[SS1]
	Blockchain-enabled	[SS26]
	Geographic-based	[SS14]
	Layered	[SS13]
	Cloud computing and the Internet of Things	[SS28]
	Distributed Blockchain-SDN based	[SS15]
	Edge Computing-based Fault Tolerant Framework	[SS12]
	SAPPARCHI	[SS21]
	Smart City	[SS17]
Smart Farm	Layered	[SS9] [SS11]
	LoRaWAN	[SS6]
	Microservices	[SS7]
	Agricultural IoT Reference Architecture (AITRA)	[SS8]

4.2.1. What are the characteristics of these IoT software system architectures?

To answer this question, we will briefly present all the IoT architecture software systems solutions, their characteristics, and how the QRs are addressed (Table 3).

Table 3. The characteristics of each IoT software systems architecture solution.

Layered Blockchain and AI-enabled [SS1]
The secure smart city infrastructure employs blockchain and AI **security** measures, IoT device authentication, and data encryption protocols. **Privacy** techniques, such as anonymization and data minimization, are also implemented while adhering to **data protection** regulations. **Real-time** threat detection and incident response mechanisms are incorporated for proactive **security**.
Layered [SS2]
The proposed healthcare monitoring framework uses wearable sensors and social networking data to monitor patients' health. It includes machine learning algorithms, cloud computing, and data analytics for real-time processing and analysis. The framework ensures **scalability**, **security**, and **high performance** using distributed processing, access control mechanisms, encryption, and secure communication protocols. It aims to improve patient care and outcomes by proactively warning patients and assisting physicians in delivering effective treatments.
Fog Computing based [SS3]
Fog-based architecture and **load balancing** can create efficient health monitoring systems. This approach uses fog nodes as intermediaries between the source and cloud and **distributes workload** across multiple nodes to maintain **efficiency under high loads**.

SDN based [SS4]
The Efficient Counter-Based DDoS Attack Detection Framework leverages SD-IoT for security and includes measures like distributed counters, access control, cryptography, and authentication. It addresses potential vulnerabilities with machine learning algorithms and incident response plans. It is a robust **security** solution for **protecting** IoT devices and networks from DDoS attacks.
Hybrid (Centralized and Distributed) [SS5]
A fog-cloud-assisted framework for the Internet of Healthcare Things (IoHT) includes a load-balancing mechanism that helps **distribute workload** across fog nodes for efficient resource utilization and system management. This mechanism plays a vital role in scaling the framework and ensuring **reliable operation** under different conditions.
LoRaWAN [SS6]
This paper presents a **cost-effective** LoRaWAN power switch architecture for smart farm applications. It includes a LoRaWAN module for wireless communication, a power switch circuit for power supply control, and user-defined inputs for customization. The architecture provides a cost-effective and enables real-time monitoring and management of farm operations.
Microservices [SS7]
The article proposes an AWS (Amazon Web Services) architecture for smart livestock monitoring. It is based on a microservices architecture pattern that includes AWS IoT services for managing and collecting sensor data, and AWS Lambda functions for processing data in real-time. The architecture provides **scalability, performance,** and flexibility for monitoring livestock in large-scale farms while incorporating **security** mechanisms such as AWS Certificate Manager and AWS Identity and Access Management (IAM) to ensure **data privacy**.
Agricultural IoT Reference Architecture (AITRA) [SS8]
The IoT architecture for agriculture prioritizes **reliability**, **security**, and **interoperability**. It supports real-time data processing and modular **scalability** for different use cases. The design emphasizes **data privacy** and **protection** against cyber-attacks, making it a secure framework for increasing productivity in the agricultural sector.
Layered [SS9]
The system uses Docker containerization for **interoperability and seamless maintenance**, open-source technologies for **cost-effectiveness**, and Debian OS for customization, **energy efficiency**, and **portability**. It has heterogeneous nodes, MQTT protocols, and AI-driven agro-weather analysis for flexibility. The user-friendly GUI layer follows a Multi-Agent System (MAS) approach to enable **scalable**, **reliable** services.
Hybrid Cloud-Fog Computing [SS10]
It is a software architecture for IoT-based healthcare systems that leverages cloud and fog computing to improve **availability**, **reliability**, **performance**, and **mobility**. The architecture uses machine learning algorithms and data analytics for predictive analysis and decision-making, providing a **scalable**, **reliable**, **high-performing** solution.
Layered [SS11]
The Smart Indoor Farms architecture enables a sustainable agricultural revolution through technological advancements. It ensures **scalability**, **security**, **privacy**, **safety**, **interoperability**, and **mobility**. The architecture **efficiently utilizes resources**, **integrates new technologies**, and adheres to regulatory standards. IoT sensors enable remote monitoring and control, while open standards facilitate **interoperability**.
Edge Computing-based Fault Tolerant Framework [SS12]
This paper presents an edge computing-based **fault-tolerant** framework for vehicular networks. It includes redundancy and distributed architecture to ensure **fault tolerance** and **availability**. The framework is designed to be scalable and flexible, providing a robust solution for enhancing fault tolerance, **portability**, and **availability** in vehicular networks.
Layered [SS13]

The proposed architecture has three layers: Data Acquisition, **Energy Efficient** Event Classification Fog, and Temporal Health Prediction with Geographic Mapping Cloud. IoT sensors collect data, and the Fog layer classifies it and optimizes transmission to the Cloud layer. The Cloud layer predicts future health states and prioritizes evacuations. The system employs intelligent data selection and dimensionality reduction techniques to contribute to **sustainability**.
Geographic-based [SS14]
SGeoL is a smart city architecture that prioritizes **Performance**, **Security**, **Interoperability**, and **Scalability**. It uses open protocols such as NGSI-LD, HTTP, and OAuth and provides developers with easy access through high-level RESTful APIs. The architecture includes SGeoL middleware and infrastructure layers, allowing efficient data storage and **flexible scalability**. SGeoL supports a range of data manipulation APIs, semantic support, storage formats, and advanced visualization.
Distributed Blockchain-SDN-based [SS15]
DistB-Condo is a **secure**, **load-balanced**, and blockchain-based model for smart condominiums. It incorporates IoT integration and software-defined networking (SDN) for centralized control, high **security**, high **reliability**, and **privacy** for IoT sensor data. It offers improved **efficiency**, advanced **security** measures, and a **balanced living experience**. Virtualization provides **load balancing**, **power saving**, and **cost-effectiveness** for smart condominiums.
Event-driven IoT [SS16]
This text describes an event-driven IoT architecture for **reliable** healthcare applications. It includes complex event processing for data analysis and various features integrated to handle data generated by sensors and devices. The architecture can improve healthcare applications by responding to **real-time** event triggers and analyzing data for better decision-making.
Smart City [SS17]
SmartGC proposes a garbage collection software architecture for smart cities that prioritizes **interoperability**, **security**, and **scalability**. It uses open standards and protocols for **interoperability**, cryptography for **protection**, and distributed elements for **scalability**. The result is a robust and **scalable** solution for efficient garbage collection and **secure** data management.
Layered [SS18]
This architecture prioritizes **security** and **privacy** through blockchain technology. It uses public and private blockchain mechanisms to secure small information units and ensure reliable and synchronized records. The fog layer enhances e-health system **performance**, allowing patients equipped with IoT sensors to transmit health data to the cloud for remote monitoring by doctors.
Cloud-based Secure Biometric [SS19]
The Health Data Management System in the healthcare cloud includes client, provider, and data center levels. Biometric authentication enhances **security**. The system uses dynamic signature features for user authentication, ensuring **energy and cost efficiency** through mobile devices. The Access Hierarchy based on user priority provides **fault tolerance**.
Multi-layered Cloud-Edge [SS20]
An architecture is proposed to secure smart healthcare systems using edge computing, which reduces the time and resources required to transfer patient data to the cloud. The framework includes encryption, access control, and identity management to ensure **data privacy** and **confidentiality**. This design enables efficient healthcare monitoring while maintaining patient data **security**.
SAPPARCHI [SS21]
SAPPARCHI is a **scalable** platform for Smart Cities that uses containerization technology to achieve horizontal **scalability**. This approach isolates application components and ensures that they can be scaled independently. The platform can efficiently handle increasing numbers of users, IoT devices, and sensors, making it a flexible solution for Smart City environments.
Layered [SS22]
The IoT system architecture uses sensor networks for **scalability**, standardized and modularized components for **maintainability**, access control and encryption for **security**, redundant components and distributed architectures for **availability**, and open and standardized protocols for **portability**.
Deep Learning based [SS23]

HealthFog system uses an ensemble deep-learning approach to diagnose heart diseases using patient data from various sensors. It ensures **data integrity**, **privacy**, **security**, **load balancing**, and **performance**. The system encrypts patient data when transmitted between IoT gateways, fog nodes, and cloud resources. It also includes a **load-balancing** mechanism and provisions for **data integrity**. Overall, it meets high-quality standards while ensuring the **security** and **privacy** of patient data and optimal **performance**.
Industrial Internet of Things (IIoT) [SS24]
The IIoT software architecture is designed to tackle industrial challenges focused on **scalability** and **interoperability**. It uses fog/edge computing to meet **real-time** and **low-latency** requirements. The architecture has four layers: Sensing/THINGS, Data Provider, Fog/Edge Computing, and Applications/Services. Each layer includes management and **security services** to address challenges such as **data security** and **real-time** capabilities. DDS middleware ensures **secure data transmission** over the Internet for a robust IIoT application **security** framework.
Distributed Edge-Cloud [SS25]
The paper proposes a secure healthcare monitoring framework integrating NDN with edge cloud technology. It includes encryption, authentication, and access control mechanisms to safeguard patient data. NDN allows **safe** data transfer, while edge cloud computing ensures real-time processing. These measures enable a **secure**, **scalable**, and **efficient** healthcare monitoring solution.
Blockchain-enabled [SS26]
The IoT architecture prioritizes **security** and **energy efficiency** using a cluster structure and distributed trust in the blockchain instead of traditional Proof of Work (POW) mechanisms. The SDN controller manages secure transactions and aims to create a resilient and **energy-efficient** IoT infrastructure by monitoring and blocking malicious nodes.
Layered Blockchain-Based SDN [SS27]
The paper proposes a secure routing architecture for SDN-enabled IoT networks that uses blockchain technology to ensure the **reliability** and **immutability** of data. The architecture includes an IoT gateway and SDN controller to facilitate **secure** communication between devices, offering a robust solution while maintaining **data integrity** and **reliability**.
Cloud computing and the Internet of Things [SS28]
The fog computing architecture integrates fog nodes with base stations, improving transmission efficiency. SDN facilitates a flexible and **scalable** fog node core network, optimizing network deployment flexibility. A layered fog computing model is introduced to address **data privacy** concerns in IoT, employing proxy virtual processors to analyze and categorize user data while protecting personal **privacy**. Two schemes for application virtual processor deployment, local and remote, offer flexibility based on the nature of data processing requirements.

4.2.2. What are the QRs identified in these IoT software system architectures?

Figure 2 shows the statistics about the QR found during the execution of this study. Each of them represents a challenge during the architecture design phase of an IoT software system. The solution for managing those QR codes is still emerging and needs to be identified and utilized as evidence of success for future developments. The graph (Figure 2) shows that security is the most present QR when designing IoT architecture software systems, influenced by performance and flexibility. The graph highlights the need for effective design features to address security concerns, such as cryptography, certificates, blockchain technology, and a layered approach to security.

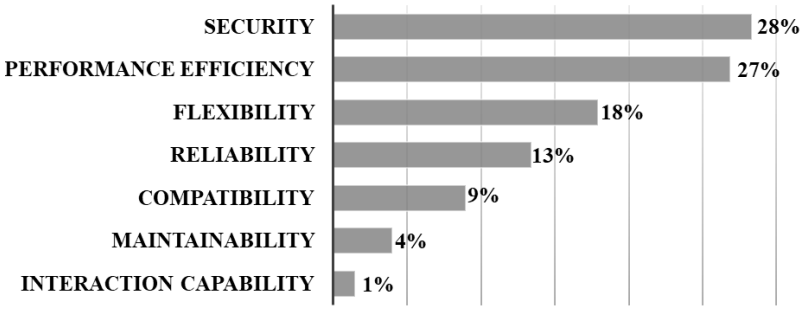

Figure 2. Statistics of the Quality Requirements found.

4.2.3. How are these QRs worked out in these IoT software system architectures?

The studied architectures have comprehensive design features (Table 5) addressing many QRs. Implementing these architectures uses technologies such as Kubernetes, PPSE, FogBus framework, blockchain technology, Docker, SDN, and others, resulting in a flexible and robust IoT software system design and implementation framework.

Table 4. Quality Requirement and Architectural Technologies.

Security		
Authenticity	- Authentication, authorization, and data access control. - IoT device authentication [Stojanov and Dobrilović 2021]. - Cloud-based biometric authentication [Shakil et al. 2020]. - OAuth protocol. - Amazon Simple Storage Service (Amazon S3).	[SS1] [SS2] [SS4] [SS8] [SS9] [SS11] [SS14] [SS15] [SS17] [SS18] [SS19] [SS20] [SS21] [SS22] [SS23] [SS24] [SS26]
Confidentiality	- Authentication, authorization, and data access control. - Searchable and encrypted edge-layer data (Privacy-Preserving Searchable Encryption - PPSE). - Blockchain technology. - Amazon Simple Storage Service (Amazon S3).	
Integrity	- FogBus framework [Tuli et al. 2019]. - Blockchain technology. - Hash.	
Resistance	- Counter-based DDoS Attack Detection (C-DAD) application for Software Defined Networks (SDN) [Bhayo et al. 2020]. - HTTP reverse proxy to protect REST services.	
Performance/Efficiency		
Resource Utilization	- Algorithm for Fog layer data selection (Data uniqueness and only important data) for transmission. - Wired communications. - Lightweight communications protocols (MQTT, HTTP, ZigBee, BLE, LoRaWAN, SigFox). - Hybrid energy design of the nodes (solar and AC power). - Network Function Virtualization (NFV). - Hypervisor (a virtual machine monitor, VMM, or virtualizer). - FogBus framework [Tuli et al. 2019]. - Load Balancing (LAB) scheme [Fan and Ansari 2020]. - Adapted Particle Swarm Optimization (APSO) [Chudhary and Sharma 2021] and Software Defined network (SDN). - Open-source technologies. - Long-Range Wide Area Network (LoRaWAN) technology. - Edge and Fog computing.	[SS3] [SS5] [SS6] [SS9] [SS13] [SS15] [SS23] [SS24]

Time Behaviour	- Data Distribution Protocol (DDS).	
Flexibility		
Adaptability	- Containerization technology (Docker). - Web-based applications. - Public/Subscriber protocols.	
Scalability	- Multi-agent system (MAS) approach. - Amazon Simple Storage Service (Amazon S3). - Publish/Subscribe to standard messaging pattern. - Distributed computing architecture. - Infrastructure-as-a-service (IaaS). - Virtualization. - Software Defined Network (SDN). - Software Defined Virtual Private Network (SD-VPN). - Amazon Web Services (AWS). - Layered Architecture with loosely coupled components. - Serverless on Edge and Microservice in Fog. - Separated Databases (complex queries and simple queries).	[SS2] [SS7] [SS9] [SS10] [SS12] [SS14] [SS21] [SS22] [SS25]
Reliability		
Availability	- Kubernetes framework (no single point of failure). - Separated Databases (complex queries and simple queries). - TCP-IP protocol on Client/Server architecture for data communication. - Reputation concept (Blockchain-based SDN-enabled network architecture) [Zeng et al. 2022]. - Software Defined Network (SDN). - Edge gateway. - Fog layer local data processing. - Data processing in the fog layer. - Multi-agent system (MAS) approach. - Fog layer in two levels: First Level – Intra mobility (User network): Nodes (Data transfer, device management, temporary local storage, notifications, and data consistency). Second Level – Extra mobility (Different domains): Smart Gateways (Local processing and database).	[SS9] [SS12] [SS15] [SS16] [SS21] [SS22] [SS27]
Fault Tolerance	- Kubernetes framework (no single point of failure).	
Compatibility		
Interoperability	- Containerization (Docker), separating the deployed software architecture from the hardware infrastructure. - RESTFul APIs. - Simple object access protocol (SOAP). - Publish/Subscribe to standard messaging pattern.	[SS9] [SS8] [SS10] [SS14] [SS22] [SS24]
Maintainability		
Modularity	- Layered Architecture with loosely coupled components.	[SS9] [SS21]
Reusability	- Containerization technology (Docker).	
Interaction Capability		
Operability	- Remote access to the deployed platform and its components (Web App and SSH).	[SS9]

4.3. What are the application domains and characteristics of their IoT software systems architectures that influence QRs?

By consolidating the data presented in the previous section, we obtain the key findings for our primary research question. The study identified 28 architectural solutions (See Table 2) corresponding to 21 QRs (See Figure 2). All architectural features were mapped to the QR (See Figure 2) and addressed as a reusable solutions catalog (See Table 4).

This study leverages the importance of the design phase when implementing an IoT software system. Overall, identifying the application domains and their characteristics is crucial in designing IoT software systems architectures that meet the specific QRs of each application domain and contribute to the successful deployment of IoT software systems in various fields, including healthcare.

5. Threats to Validity

Threats to validity in empirical studies always demand consideration. The first consideration is about the selection of the initial works. We used clear and objective inclusion and exclusion criteria (See Table 1) to compose the first papers from the Scopus database. Moreover, two related works (See section 2) were selected and used to refine our search procedure, reducing this threat and increasing reliability. Primary sources were identified, and inclusion and exclusion criteria, full reading, and snowballing techniques were performed for a concreted conclusive outcome.

Two researchers identified the articles to mitigate potential biases in the chosen papers and the interpretation bias among researchers, while another conducted a comprehensive review of the final set. The research protocol is designed to enhance the traceability of its data.

Finally, taxonomy bias was mitigated by referencing the works with the same objectives and theme (See section 2). The structure of these studies was considered and reused to map the goals of this work. Overall, every effort was made to conduct a secondary, well-conducted survey, aiming to reduce the threats to validity and increase the reliability and validity of the results.

6. Conclusion and Future Works

The development of IoT software systems architectures has been experiencing substantial growth, driven by increasing demand, adaptation requirements, and emerging challenges. Therefore, reusing successful cases in the software engineering process is crucial for supporting decision-making in designing IoT software systems architecture strategies. So, we collect data from primary sources to maintain the database of challenges and solutions, focusing on QRs. A list of IoT domain areas was introduced, an IoT architectures classification was demonstrated, and QRs architectural solutions were illustrated.

Based on what has been exposed, many software systems with cloud-based processing are observed to meet the requirements of energy efficiency and performance criteria, for example. The use of layers in architectures for a separation of concerns and good maintainability also remains a common decision in the design of the solutions. Security, leveraged by performance efficiency and flexibility, is the QR most considered

when designing IoT software system architectures. Blockchain, cryptography, and SDN are very useful for mitigating these challenges.

More solutions and challenges must be added to this database for future research as continuous work. Furthermore, the data from related works and this study must be filtered and structured for application in a decision tree. This tool will aid in making decisions during the design phase of IoT software systems architecture development.

Acknowledgments

This work is partially supported by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior -Brasil (CAPES) -Finance Code 001 and by CNPq. Prof. Travasssos is a CNPq Researcher (305701/2022-3) and CNE FAPERJ (E-26/201.170/2021, which support this research.

References

Alreshidi, A. and Ahmad, A. (2019). Architecting software for the Internet of Things-based systems. Future Internet, v. 11, n. 7, p. 153.

Atzori, L., Iera, A. and Morabito, G. (2010). The Internet of Things: A survey. Computer Networks, v. 54, n. 15, p. 2787–2805.

Davami, F., Adabi, S., Rezaee, A. and Rahmani, A. M. (2021). Fog-based architecture for scheduling multiple workflows with high availability requirements. Computing, v. 104, n. 1, p. 169–208.

Fan, Q. and Ansari, N. (2020). Towards workload balancing in fog computing empowered IoT. IEEE Transactions on Network Science and Engineering, v. 7, n. 1, p. 253–262.

Gubbi, J., Buyya, R., Marusic, S. and Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, v. 29, n. 7, p. 1645–1660.

ISO/IEC-25010 (2023). ISO/IEC 25010. ISO 25010—Systems and Software Quality Requirements and Evaluation (SQuaRE)—System and software quality models. ISO.org. ISO/IEC Fdis 25010:2023. https://www.iso.org/obp/ui/en/#iso:std:iso-iec:25010:ed-2:v1:en. v. 2010, 2023.

Kuhrmann, M., Méndez, D. and Daneva, M. (2017). On the pragmatic design of literature studies in software engineering: an experience-based guideline. ESE, v. 22, n. 6, p. 2852–2891.

Mirani, A. A., Velasco-Hernandez, G., Awasthi, A. and Walsh, J. (2022). Key Challenges and Emerging Technologies in Industrial IoT architectures: A review. Sensors, v. 22, n. 15, p. 5836.

Motta, R. C., Silva, V. and Travassos, G. H. (2019). Towards a more in-depth understanding of the IoT Paradigm and its challenges. JSERD, v. 7, p. 3.

Nakagawa, E. and Antonio, P. [Eds.] (2023). Reference architectures for critical domains: Industrial Uses and Impacts. 1. ed. Springer Cham.

Razzaq, A. (20 oct 2020). A Systematic Review of software architectures for IoT systems and future direction to adopting a microservices architecture. SN Computer Science, v. 1, n. 6.

Tuli, S., Mahmud, R. and Buyya, R. (2019). FogBus: a blockchain-based lightweight framework for edge and fog computing. JSS, v. 154, p. 22–36.

Wöhlin, C. (2014). Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering. EASE 14. p. 1–10.

Appendix A

Selected Studies

SS1	Ahmed, I., Zhang, Y., Jeon, G., et al. (2022). A blockchain- and artificial intelligence-enabled smart IoT framework for sustainable city. International Journal of Intelligent Systems, v. 37, n. 9, p. 6493–6507.
SS2	Ali, F., El–Sappagh, S., Islam, S. M. R., et al. (2021). An intelligent healthcare monitoring framework using wearable sensors and social networking data. Future Generation Computer Systems, v. 114, p. 23–43.
SS3	Asghar, A., Abbas, A., Khattak, H. A. and Khan, S. U. (2021). FOG-based architecture and load balancing methodology for health monitoring systems. IEEE Access, v. 9, p. 96189–96200.
SS4	Bhayo, J., Hameed, S. and Shah, S. A. (2020). An efficient Counter-Based DDOS attack detection framework leveraging software-defined IoT (SD-IoT). IEEE Access, v. 8, p. 221612–221631.
SS5	Chudhary, R. and Sharma, S. (2021). Fog-cloud assisted framework for Heterogeneous Internet of Healthcare Things. Procedia Computer Science, v. 184, p. 194–201.
SS6	Da Silveira, J. D. F., Da S Veloso, A. F., Reis, J. V. D., Soares, A. and Rabelo, R. a. L. (17 oct 2021). A new Low-Cost LORAWAN power switch for smart farm applications—2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC).
SS7	Dineva, K. and Atanasova, T. (2021). Design of scalable IoT architecture based on AWS for smart livestock. Animals, v. 11, n. 9, p. 2697.
SS8	El-Basioni, B. M. M. and El-Kader, S. M. A. (2020). Laying the foundations for an IoT reference architecture for the agricultural application domain. IEEE Access, v. 8, p. 190194–190230.
SS9	Faid, A., Sadik, M. and Sabir, E. (2021). An agile AI and IoT-Augmented Smart Farming: a Cost-Effective cognitive weather station. Agriculture, v. 12, n. 1, p. 35.
SS10	Hajvali, M., Adabi, S., Rezaee, A. and Hosseinzadeh, M. (2021). Software architecture for IoT-based healthcare systems with cloud/fog service model. Cluster Computing, v. 25, n. 1, p. 91–118.
SS11	Hati, A. J. and Singh, R. R. (2021). Smart Indoor Farms: Leveraging technological advancements to power a sustainable agricultural revolution. AgriEngineering, v. 3, n. 4, p. 728–767.
SS12	Javed, A., Malhi, A. and Främling, K. (2020). Edge Computing-based Fault-Tolerant Framework: A Case Study on Vehicular Networks. International Wireless Communications and Mobile Computing (IWCMC),
SS13	Kaur, A., Sahil, S. and Sood, S. K. (2022). Cloud-FOG assisted Energy Efficient Architectural Paradigm for disaster evacuation. Information Systems, v. 107, p. 101732.
SS14	Pereira, J., Batista, T., Cavalcante, E., et al. (2022). A platform for integrating heterogeneous data and developing smart city applications. Future Generation Computer Systems, v. 128, p. 552–566.
SS15	Rahman, A., Islam, Md. J., Rahman, Z., et al. (2020). DISTB-ConDo: Distributed Blockchain-Based IoT-SDN model for smart condominium. IEEE Access, v. 8, p. 209594–209609.
SS16	Rahmani, A. M., Babaei, Z. and Souri, A. (2020). Event-driven IoT architecture for data analysis of reliable healthcare applications using complex event processing. Cluster Computing, v. 24, n. 2, p. 1347–1360.

SS17	Ramalho, M. S., Rossetti, R. J. F., Cacho, N. and Souza, A. S. (2020). SmartGC: a software architecture for garbage collection in smart cities. International Journal of Bio-inspired Computation, v. 16, n. 2, p. 79.
SS18	Safdar, Z., Farid, S., Qadir, M., et al. (2020). A novel architecture for the Internet of Things based E-Health systems. Journal of Medical Imaging and Health Informatics, v. 10, n. 10, p. 2378–2388.
SS19	Shakil, K. A., Zareen, F. J., Alam, M. and Jabin, S. (2020). BAMHealthCloud: A biometric authentication and data management system for healthcare data in the cloud. Journal of King Saud University - Computer and Information Sciences, v. 32, n. 1, p. 57–64.
SS20	Singh, A. and Chatterjee, K. (2021). Securing smart healthcare system with edge computing. Computers & Security, v. 108, p. 102353.
SS21	Souza, A., Cacho, N., Batista, T. and Ranjan, R. (2022). SAPPARCHI: an Osmotic Platform to Execute Scalable Applications on Smart City Environments. IEEE 15th International Conference on Cloud Computing (CLOUD).
SS22	Stojanov, Ž. and Dobrilović, D. (2021). Software architecture quality attributes of a layered sensor-based IoT system (short paper). Workshop Information Technologies: Algorithms, Models, Systems (ITAMS), p. 66–74.
SS23	Tuli, S., Basumatary, N., Gill, S. S., et al. (2020). HealthFog: An ensemble deep learning-based Smart Healthcare System for Automatic Diagnosis of Heart Diseases in integrated IoT and fog computing environments. Future Generation Computer Systems, v. 104, p. 187–200.
SS24	Ungurean, I. and Gaitan, N. C. (2020). A software architecture for the Industrial Internet of Things—A Conceptual model. Sensors, v. 20, n. 19, p. 5603.
SS25	Wang, X. and Cai, S. (2020). Secure healthcare monitoring framework integrating NDN-based IoT with edge cloud. Future Generation Computer Systems, v. 112, p. 320–329.
SS26	Yazdinejad, A., Parizi, R. M., Dehghantanha, A., Zhang, Q. and Choo, K. R. (2020). An Energy-Efficient SDN controller architecture for IoT networks with Blockchain-Based security. IEEE Transactions on Services Computing, v. 13, n. 4, p. 625–638.
SS27	Zeng, Z., Zhang, X. and Xia, Z. (2022). Intelligent Blockchain-Based secure routing for multidomain SDN-Enabled IoT networks. Wireless Communications and Mobile Computing, v. 2022, p. 1–10.
SS28	Zhang, C. (2020). Design and application of fog computing and Internet of Things service platform for smart city. Future Generation Computer Systems, v. 112, p. 630–640.