applied sciences

MDPI

Article

System Architecture Design of IoT-Based Smart Cities

Bedir Tekinerdogan [1,*], Ömer Köksal [2] and Turgay Çelik [3]

[1] Information Technology Group, Wageningen University & Research, 6708 PB Wageningen, The Netherlands
[2] Aselsan, Ankara 06172, Turkey
[3] Turkish Aerospace Industries, Ankara 06980, Turkey
[*] Correspondence: bedir.tekinerdogan@wur.nlT

Abstract: The integration of Internet of Things (IoT) networks into smart cities is crucial to enhance the efficiency of city operations and services. Designing a smart city architecture that can adapt to the constantly changing functional and quality requirements of city services is essential. However, critical decisions must be made during this process, such as selecting communication protocols, ensuring security and safety, optimizing time performance, and processing data capacity. To address these challenges, this paper proposes a systematic approach to guide the system architecture design of IoT-based smart cities. The approach starts with feature-driven domain analysis to model smart city requirements, followed by the design of a reference architecture for IoT-based smart cities. The architecture is modeled using selected architectural views, while considering key stakeholders and their concerns. Additionally, this paper presents valuable insights into lessons learned and challenges encountered during the process of creating IoT-based smart cities. This information can assist practitioners in developing such smart cities and pave the way for future research in this field. By following this proposed approach, smart city architects can design a robust and adaptable system architecture that can meet the evolving needs of smart city services.

Keywords: smart city; internet of things; system architecture

check for updates

Citation: Tekinerdogan, B.; Köksal, Ö.; Çelik, T. System Architecture Design of IoT-Based Smart Cities. *Appl. Sci.* **2023**, *13*, 4173. https://doi.org/10.3390/app13074173

Academic Editor: Marco Mesiti

Received: 20 February 2023
Revised: 18 March 2023
Accepted: 22 March 2023
Published: 24 March 2023

1. Introduction

The development of smart cities is a rapidly growing trend in today's urban environments. A smart city is a city that uses advanced technology, data analytics, and the internet of things (IoT) to optimize the efficiency of city operations and services [1–5]. The goal of a smart city is to enhance quality of life for citizens, attract businesses and investors, and create a sustainable and resilient environment. Smart cities typically have a wide range of interconnected systems and devices that can be managed remotely, including transportation systems, energy systems, public safety, healthcare, and waste management. They also rely on data and analytics to make informed decisions and improve the city's overall performance [6].

Designing a smart city is a complex task requiring the integration of various technologies and systems, including IoT devices, communication protocols, and data processing and management systems [7–10]. A smart city can also be characterized as a system of systems, and it has to cope with several obstacles [11]. The design of a smart city architecture must be scalable and dynamic in order to cope with the continuously evolving functional and quality requirements for city services. Different important design decisions must be made, including the selection of communication protocols, security levels, safety levels, time performance, and data processing capacity.

To address these challenges, in this paper, we propose a systematic method for guiding the system architecture design of IoT-based smart cities. First, we adopt a feature-driven domain analysis to model the smart city requirements, followed by the design of a reference architecture for IoT-based smart cities. The proposed architecture includes a layered structure, with the physical layer consisting of IoT devices, the network layer comprising communication protocols, and the application layer comprising various smart city services. The architecture also includes a management layer for monitoring and controlling the overall system.

We provide detailed descriptions of each layer, including the challenges and potential solutions for their implementation. Furthermore, we discuss the proposed architecture's security considerations and potential threats. Additionally, we present a case study of a smart city to demonstrate the practical application of the proposed architecture. The proposed architecture can serve as a blueprint for developing smart cities, enabling the integration of various IoT devices and services to benefit citizens and city management.

The novelty and the contribution of this paper are as follows:

- A systematic method is proposed for guiding the system architecture design of IoT-based smart cities;
- The method includes a feature-driven domain analysis to model the smart city requirements and the design of a reference architecture for IoT-based smart cities;
- The paper presents a case study of a smart city to demonstrate the practical application of the proposed architecture;
- The proposed reference architecture can serve as a blueprint for developing smart cities, enabling the integration of various IoT devices and services to benefit citizens and city management.

The remainder of this article is organized as follows. Section 2 provides the background on the key elements, including the internet of things and the challenges of designing smart cities. Section 3 describes the adopted research method and the feature-driven domain analysis approach used to model the smart city requirements. Section 4 presents a case study of a smart city to demonstrate the practical application of the proposed architecture. In Section 5, we present the IoT architecture of the smart city and discuss the key components of each layer. Section 6 presents the related work and compares our proposed architecture with existing solutions. Finally, Section 7 concludes the paper and discusses future research directions.

2. Background

2.1. Smart City

A smart city is a concept that refers to the use of technology and data to improve the quality of life, efficiency, and sustainability of urban areas. Smart cities aim to optimize the use of resources, reduce waste and pollution, and improve the overall livability of urban areas by making them more responsive to the needs of citizens. Smart cities leverage technology to improve the services they offer to citizens, and there are many different applications of smart city technology, depending on the specific needs and priorities of a particular city:

- Intelligent transportation systems which can optimize traffic flow, reduce congestion, and improve public transportation;
- Smart building management systems that can reduce energy consumption and improve the overall comfort of citizens;
- Smart lighting systems that can adjust lighting levels based on the presence of people and vehicles;
- Smart parking systems that can help drivers find available parking spaces quickly and easily;
- Smart waste management systems that can optimize the collection and disposal of waste;
- Smart water management systems that can improve the quality and distribution of water.

As stated before, one of the key components of a smart city is the Internet of Things (IoT) technology. In general, the goal of a smart city is to use technology and data to improve the quality of life, efficiency, and sustainability of urban areas. By making cities more responsive to the needs of citizens and by using advanced communication and IoT technology to optimize the use of resources, smart cities can help to create more livable, sustainable, and resilient communities.

2.2. Internet of Things

The Internet of Things (IoT) refers to the interconnectedness of devices through the internet. Historically, the internet was primarily used for connecting computers, requiring human interaction and monitoring. However, the IoT expands upon this concept by adding a new dimension to current information and communication technologies (ICTs), enabling 'anytime' and 'anyplace' connectivity for 'anything' by connecting physical and virtual objects in the real world.

IoT encompasses various technologies, including wireless sensor networks, machine-to-machine communication, mobile computing, ubiquitous computing, and embedded systems. The International Telecommunication Union (ITU) defines IoT as "the network of physical objects or things" embedded with electronics, software, sensors, and network connectivity, which enables these objects to collect and exchange data [10].

IoT architecture is typically a layered structure, grouping modules offering cohesive services. Several reference architectures have been proposed in the literature, with common layers including the device, network, and application layer [1,12–14].

The Internet of Things (IoT) architecture can be represented as a layered structure consisting of the following layers: device, network, session, application, business, management, and security (Figure 1). The device layer comprises sensors and physical devices that collect and identify data and information generated by sensors and physical devices. This data is then passed on to the network layer, which provides networking connectivity and transport capabilities, allowing for the secure transmission of data to the session layer. The session layer is responsible for service management, including setting up and breaking down associations between IoT connection points. Standards and protocols offered by different organizations can be used in this layer, most utilizing the Transmission Control Protocol (TCP) or the User Datagram Protocol (UDP) for transport. The application layer manages the IoT system using data from the session layer and contains IoT services such as smart farming, smart cities, and smart homes. The business layer defines business logic and workflows, manages all IoT systems, services, and applications within the domain, and analyzes data from the application layer to build required business models and strategies. Finally, the security layer provides security functionality, and the management layer supports capabilities such as device management, local network topology management, and traffic and congestion management. In fact, security mechanisms also need to be implemented at various layers of the architecture. For example, at the network layer, secure communication protocols can be implemented to ensure the confidentiality and integrity of data transmitted over the network. At the session layer, encryption techniques can be used to protect sensitive data, and access control mechanisms can be implemented to ensure that only authorized users are able to access the data. At the application layer, role-based access control can be used to ensure that users have access only to the data and services that they require for their specific roles. In addition to these measures, ongoing security monitoring and evaluation are crucial to identify and address security risks and vulnerabilities over time. This can include implementing regular security assessments and testing procedures, as well as developing incident response plans to ensure that security breaches are detected and addressed in a timely manner.

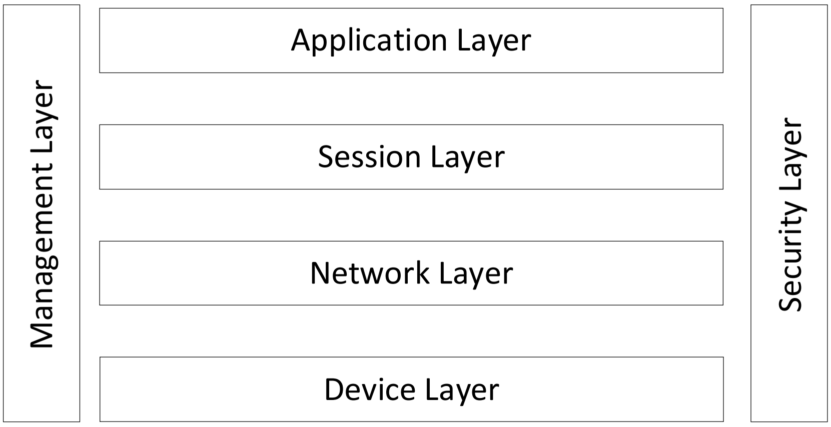

Figure 1. IoT Reference Architecture.

A critical aspect of IoT systems is interoperability which enables different devices (e.g., sensors) and systems to communicate and exchange data effectively. The IoT reference architecture addresses interoperability in several ways, including the use of standardized protocols, a layered approach to system design, and the incorporation of middleware to enable data translation and communication between different systems. First, the IoT reference architecture relies on standardized protocols for data exchange, such as MQTT, CoAP, and HTTP. Standardization of protocols ensures that different devices and systems can communicate with each other, regardless of the manufacturer or the technology used. This promotes interoperability by ensuring that data can be exchanged seamlessly between different devices and systems. Second, the IoT reference architecture uses a layered approach to system design, with each layer providing specific functionality. This layered approach enables system designers to isolate and manage different aspects of the system separately, which makes it easier to ensure interoperability between different layers. Finally, the middleware layer is a critical component of the IoT reference architecture, as it enables data translation and communication between different systems. The middleware layer acts as a bridge between the perception and application layers, translating data between different formats and protocols. This enables different devices and systems to communicate and exchange data effectively, regardless of their location or the network they are on.

2.3. Architecture Design

Software architecture for a program or computing system is the foundation of its design and development. It encompasses the structure, elements, and relationships that make up the system and is one of the key artifacts in the entire software development life cycle. Software architecture design plays a crucial role in addressing the challenges of developing large-scale and complex software systems.

Various techniques and strategies have been developed to improve software architecture design in recent years. One important aspect of this is considering the concerns of stakeholders, individuals, teams, or organizations with an interest in or impact on the system. These concerns shape the early design decisions made by the architect and are often modeled using different "architectural views" that represent the system's elements and relations with specific stakeholder concerns.

One approach to modeling architectural views is the "Views and Beyond" (V and B) approach, which differentiates three categories of viewpoints or styles: module, component and connector, and allocation styles [15]. The module view category is used for documenting the system's principal units of implementation, the component and connector category for documenting the system's units of execution, and the deployment view category for documenting the relationships between the system's software and its development and execution environments.

A software architecture that addresses the concerns of specific stakeholders is known as concrete architecture. This architecture defines the boundaries and constraints for implementation and is used to analyze risks, balance trade-offs, plan implementation projects, and allocate tasks. Concrete architectures are specific implementations of reference architectures, generic designs derived from past knowledge and experiences. These reference architectures can be used descriptively to capture the essence of existing architectures or prescriptively to guide the development of new ones.

3. Research Method

The research method adopted for this study includes three main steps, namely domain analysis, architecture design, and multi-case study research (Figure 2). The description of each step is as follows.

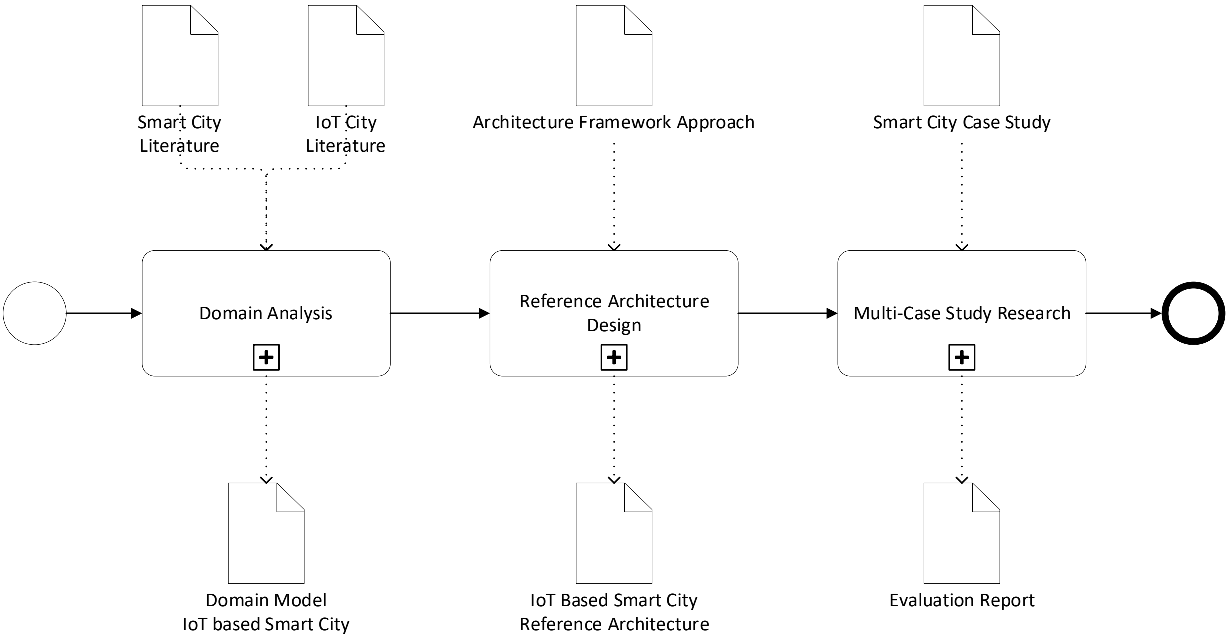
Smart City Literature
IoT City Literature
Architecture Framework Approach
Smart City Case Study
Domain Analysis
Reference Architecture Design
Multi-Case Study Research
Domain Model IoT based Smart City
IoT Based Smart City Reference Architecture
Evaluation Report

Figure 2. Adopted Research Methodology.

Domain Analysis: The first step in the research process is to conduct a thorough domain analysis that involves a review of existing literature on the topic of IoT and smart cities, as well as an analysis of existing smart city projects and initiatives. This step aims to understand the field's current state, identify key challenges and opportunities, and identify areas for further research.

Architecture Design: Based on the findings from the domain analysis, the next step is to design an architecture for the IoT in smart cities. This architecture should consider the specific requirements and constraints of the smart city environment and should be designed to support the development and deployment of IoT-based solutions. Furthermore, the architecture should also be flexible enough to adapt to future technological advancements and changing requirements.

Multi-Case Study Research: The final step in the research process is to conduct multi-case study research. This involves selecting several smart cities and conducting in-depth case studies of their IoT projects and initiatives. This step aims to gain a detailed understanding of the challenges and opportunities faced by these cities, as well as to identify best practices and lessons learned. In addition, the case studies will be used to validate the proposed architecture and to identify areas for further research.

4. Domain Analysis

The first step of our research method involves domain analysis of IoT-based smart city engineering. Domain analysis is generally defined as an approach to understanding the problem space and context of a particular field or industry. It involves gathering and analyzing information about user needs, business processes, and technical constraints in a specific domain to help inform the design and development of software solutions. In essence, it consists of domain scoping and domain modeling.

Domain scoping is the first step of domain analysis and involves defining the boundaries of the problem space and identifying what specific aspects of the domain are relevant to the software development project at hand. It is used to determine the scope and goals of the domain analysis and to ensure that the investigation is focused on the domain's most important and relevant aspects. The scope of our research is the domain of smart cities developed using IoT. As such, both domains and, in particular, the combination of both domains is explored.

Domain modeling is the next step of domain analysis and involves creating conceptual representations of real-world entities, processes, and relationships within the defined scope of the problem space. It is used to gain a deeper understanding of the domain and to identify potential solutions for the system development project. Feature modeling is a technique used as part of domain modeling in software engineering to represent a software system's product requirements and variability. It is a graphical representation that captures the common and variant features and the relationships between features. Feature modeling plays a crucial role in capturing the product requirements and variability of the system. In feature modeling, features are defined as distinct and recognizable characteristics or capabilities of a system. They can be used to represent functional and non-functional requirements, as well as the variability of the system. The adopted relations in feature modeling define the relationships between features, such as mandatory, optional, and alternative relationships. The following are the most common relations:

- Mandatory: A feature must be included in the final product;
- Optional: A feature may be included in the final product, but it is not required;
- Alternative: Only one of the features in this group can be included in the final product;
- Or: At least one feature in this group must be included in the final product;
- Exclusive Or: Only one of the features in this group can be included in the final product, but the others are not allowed.

By using these relations, feature modeling can capture the hierarchical structure of features and the interdependencies between features, which can help ensure that the final solution meets stakeholders' requirements and is flexible and scalable. The resulting feature model for IoT-based smart city systems is shown in Figure 3. A detailed explanation of the specific features and the interconnection is provided in our earlier work [16,17].

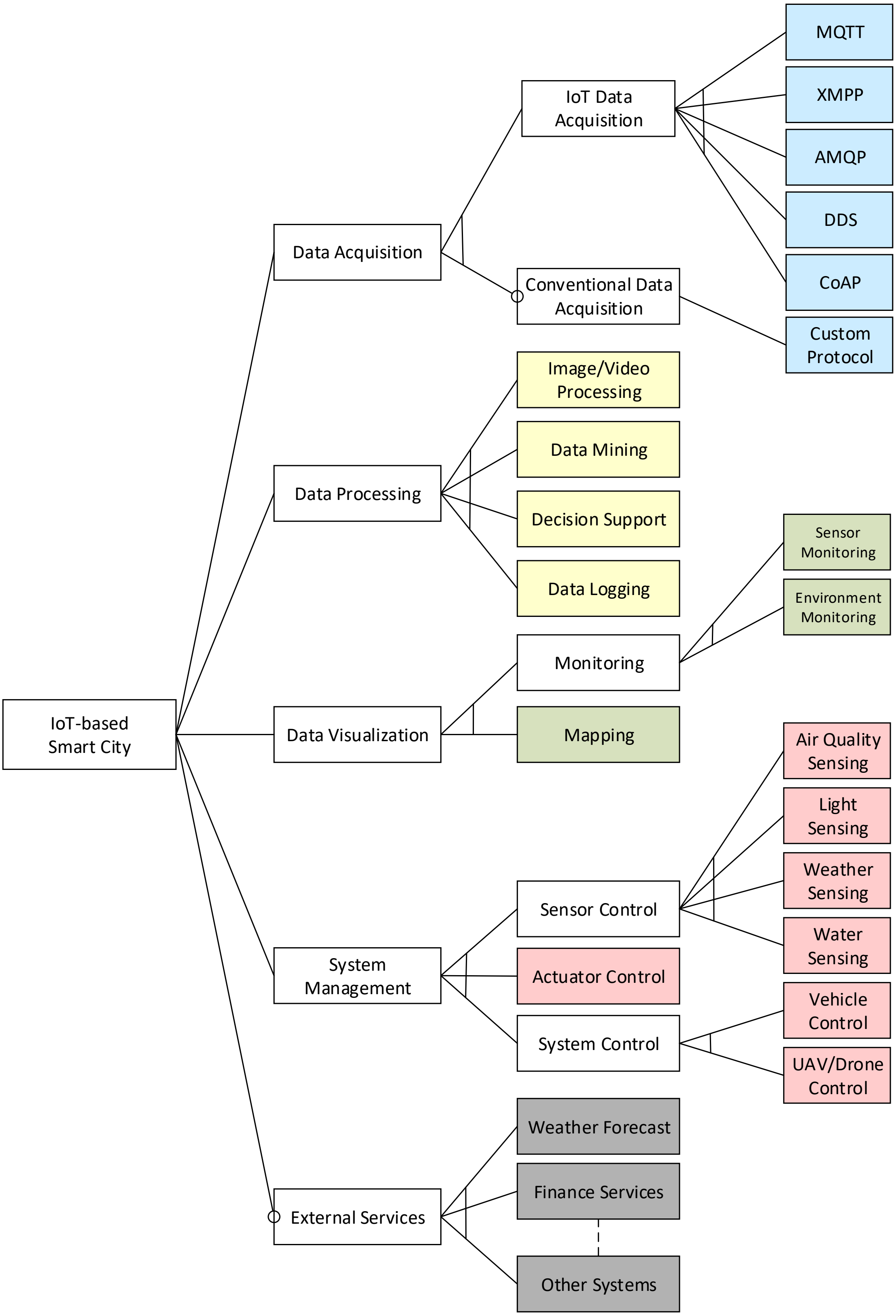

Figure 3. Feature Model for IoT-based Smart Cities.

5. Reference Architecture Design

5.1. Architecture Framework and Viewpoint Selection

The selection of an architecture framework and viewpoints is a crucial step in the software development process as it forms the foundation for the design and implementation of a system. The choice of a framework and viewpoints should be based on the specific needs and concerns of the stakeholders involved in the project.

When selecting an architecture framework, it is important to consider the overall goals and objectives of the system, as well as the constraints and limitations imposed by the development environment. In addition, the framework should provide a structure and methodology for modeling and analyzing the system. Finally, it should be supported by tools and resources to aid the design and implementation process.

Once a framework has been chosen, selecting viewpoints is the next step. Viewpoints represent the stakeholders' different concerns and should be chosen based on their relevance to the system being developed. For example, the Views and Beyond (V and B) approach distinguishes three different viewpoints categories: the module view, component and connector view, and deployment view. These viewpoints document the system's principal units of implementation, its units of execution, and the relationships between the system's software and its development and execution environments.

It is important to note that selecting a framework and viewpoints is not a one-time process but rather an ongoing process that may change over time as the system evolves. As the system is developed, new concerns may arise, and it may be necessary to modify or add new viewpoints to the architecture to address these concerns.

Decomposition, layered, and deployment viewpoints have been selected for the smart city case study for a few reasons. First, the decomposition viewpoint is useful for breaking down a complex system into smaller, more manageable parts. It helps to identify the key components of a system and how they interact with one another. In the case of a smart city, decomposition can be used to identify the various subsystems that make up the city, such as transportation, energy, and communication systems. This can aid in understanding how different parts of the city interact and how to optimize their performance.

The layered viewpoint, on the other hand, is useful for organizing the different components of a system into logical layers. This helps to separate concerns and make the system more modular. For example, in the case of a smart city, layering can be used to separate the different levels of control and management, such as the city-wide control system, the district control system, and the local control system. This can aid in understanding how control and management flow through the city and how to optimize performance.

The deployment viewpoint, finally, is useful for understanding how a system is deployed in the real world. It helps to identify the physical components of a system and how they are connected. For example, in the case of a smart city, deployment can be used to identify the different physical infrastructure that makes up the city, such as roads, buildings, and communication networks. This can aid in understanding how the city is physically constructed and how to optimize its performance.

In the following subsections, we describe the architectural views for each of these viewpoints.

5.2. Decomposition View

The decomposition view for the IoT-based smart city system architecture is shown in Figure 4. The decomposition view displays the identified implementation modules. The figure uses the colors adopted in the feature diagram of Figure 3. This way, traceability links between features and the corresponding modules in the decomposition view are defined. For a given system, different features and or modules will be selected. The decomposition view here shows thus all possible modules that could be chosen.

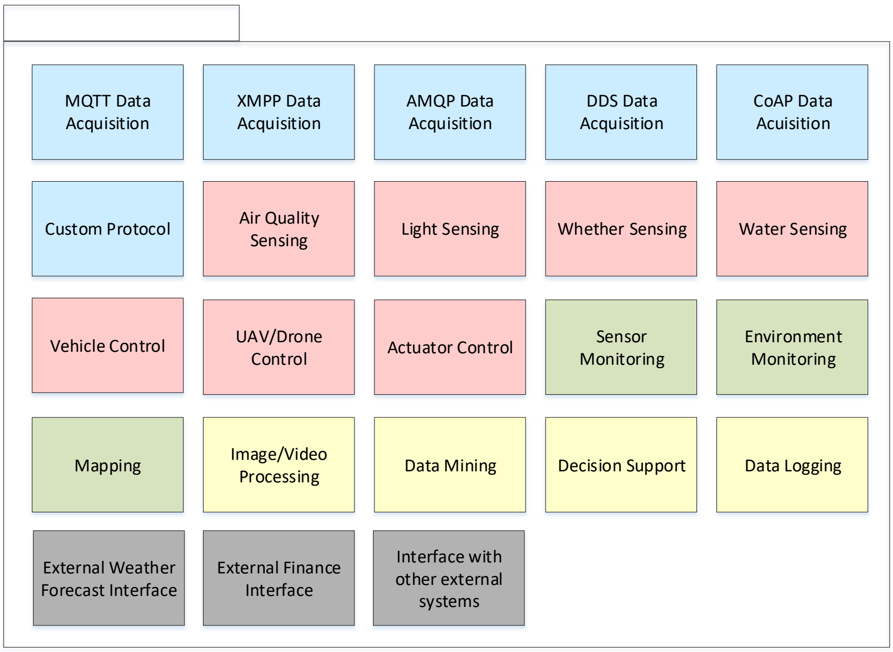

Figure 4. Reference Decomposition View for IoT-Based Smart City Inf. Man. Systems.

5.3. Layered View

The layered view is shown in Figure 5. As seen from the figure, the layered view adopts the reference architecture for IoT systems published in the literature. However, for each layer, the possible modules are provided. Again, this view is a reference view and can be used to develop an application view. This means, yet again, that some modules might be selected. However, the layers themselves are common for all systems.

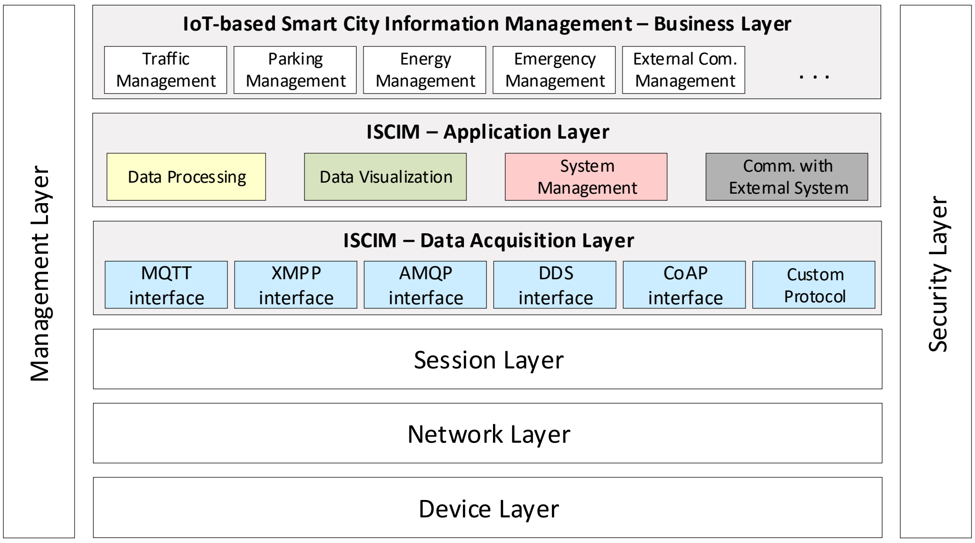

Figure 5. Reference Layered View for IoT-Based Smart City Inf. Man. Systems.

5.4. Deployment View

The previously defined decomposition and layered views focus on modeling the software modules of IoT-based smart city systems. The deployment view builds upon these views by illustrating the assignment of software modules to the hardware of a computing platform. Figure 6 displays the deployment view of these systems. The data processing module is deployed on the central cloud server and client nodes, while other nodes are dedicated to sensors, actuators, and cameras. The primary sensors in UAVs/drones and satellites are cameras, while vehicles may have their sensors, actuators, and cameras, which can be assigned to different nodes.

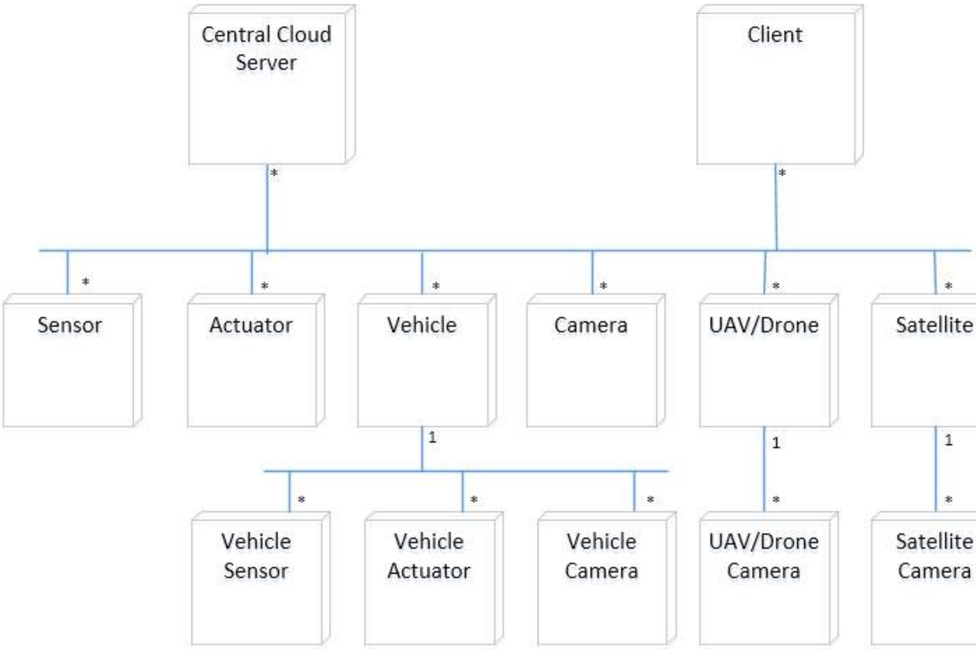

Figure 6. Reference Deployment View for IoT-Based Smart City Inf. Man. Systems (1 and * stands for one-to-many relation).

6. Case Study Research

In this section, we define the case study protocol and the selected case study that will be used to illustrate the problem statement and the approach. The case study protocol is described in Section 6.1, the case study is described in Section 6.2, and the architectural views for the case study are in Section 6.3.

6.1. Case Study Protocol Description

The primary objective of the case studies is to evaluate the impact of the developed architecture design method for IoT-based Smart Cities. Therefore, the case study research protocol defined by Runeson and Höst [16] has been applied. Based on this, the indicated five steps have been followed: (1) case study design; (2) preparation for data collection; (3) execution with data collection on the studied case; (4) analysis of collected data; (5) reporting.

Table 1 shows the case study design elements. The case study research has been applied to the smart charging case study. The case study's goal is to evaluate the effectiveness and practicality of the approach. The research questions are defined accordingly, as shown in the table. For the adopted background and sources in the case study research, official design documents were used, and interviews were held with project managers and system architects. A qualitative data analysis approach has been used. In addition, indirect data analysis has been used by analyzing the requirements provided in the literature, applying the method, and comparing the results of the method with the existing architecture.

Table 1. Case Study Design.

Case Study Design Activity	Case Study Element
Goal	Assessing the effectiveness of the method Assessing the practicality of the method
Research Questions	RQ1. To what extent does the method support the architecture design of the DDS-Based Smart City IoT architecture? RQ2. How practical is the method for deriving DDS-Based Smart City IoT architecture?
Background and source	Official requirement documents Project managers and system architects
Data Collection	Indirect data collection and direct data collection through semi-structured interviews (mix of open and closed questions)
Data Analysis	Qualitative data analysis

6.2. Case Study-Smart Charging

The previous section has described the reference architecture for IoT-based smart cities. A family feature diagram, the reference decomposition view, the reference layered view, and a reference deployment view has been provided. In this section, we provide a case study on smart

The proposed smart EV charging system is designed to provide electric vehicle (EV) owners with a convenient, reliable, and sustainable solution for charging their vehicles. The system's design focuses on placing EV charging stations in key public locations, such as parking lots and shopping centers, that are easily accessible to EV owners. This will help EV owners to charge their vehicles at a convenient location and at a time that is most suitable for them.

The EV charging stations in the system will be connected to a centralized management system, including each physical charging station's virtual replica or digital twin (Figure 7). The virtual representation will reflect the current state, such as usage and power status, of the corresponding physical station in real-time. This will allow the central management system to monitor and control the charging stations remotely, ensuring they operate efficiently and effectively.

The "EV Charging Central Management" application will maintain the central management system, which will interact with other services, such as payment and database. These services will help manage the financial transactions associated with EV charging, such as payments and billing. Additionally, the central management system will also communicate with external systems, such as banking and city electricity management. This will enable the system to integrate with other systems and provide a more comprehensive service to EV owners.

EV owners can access the central management system through a mobile or web application, which will allow them to locate available charging stations, view real-time charging status and pricing information, and initiate and pay for charging sessions. This will make charging EVs more efficient and user-friendly, promoting greater adoption of electric vehicles. The mobile or web application will also allow EV owners to schedule charging sessions in advance, ensuring the charging station is available when needed. Furthermore, the app will provide real-time notifications on the charging status so EV owners can plan their trips accordingly.

The proposed smart EV charging system aims thus to provide a convenient, reliable, and sustainable solution for charging EVs. Furthermore, integrating the charging stations with a centralized management system will enable efficient and effective management of the charging stations, making charging EVs more user-friendly for EV owners. This will help promote greater electric vehicle adoption and contribute to a more sustainable future.

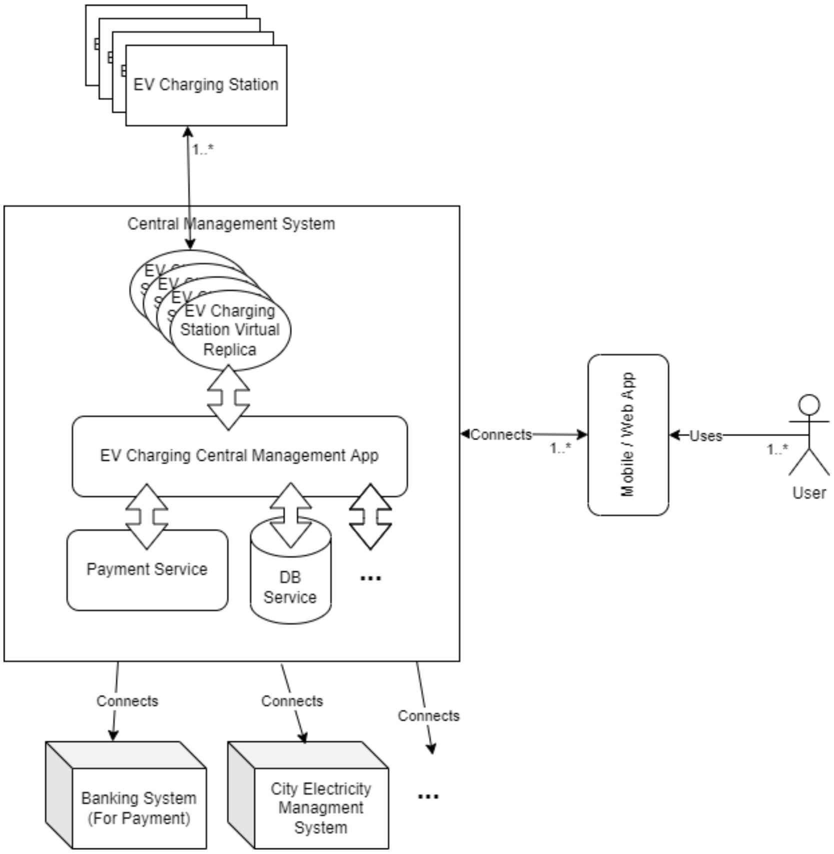
EV Charging Station
1..*
Central Management System
EV Charging Station Virtual Replica
EV Charging Central Management App
Payment Service
DB Service
...
Connects
1..*
Mobile / Web App
Uses
1..*
User
Connects
Connects
Connects
Banking System (For Payment)
City Electricity Managment System
...

Figure 7. Conceptual diagram of Smart Charging System (1 and * stands for one-to-many relation).

While the proposed smart EV charging system has several benefits, it also comes with deployment, infrastructure, and maintenance costs. The deployment cost involves installing the charging stations at various public locations, including parking lots and shopping centers, where EV owners can easily access them. The infrastructure cost includes setting up a centralized management system that can monitor and control the charging stations remotely. This system requires a reliable and robust network infrastructure to ensure the charging stations' real-time data are transmitted and processed efficiently.

Additionally, maintaining the charging stations and the central management system requires ongoing upkeep, such as regular software updates, technical support, and hardware repairs. These costs can be offset by revenue generated from charging fees and other value-added services, such as advertising and data analytics.

6.3. Architecture Viewpoints

We can now define the application views for the smart charging system based on the reference architecture. These views include the application decomposition view (Figure 8), the application layered view (Figure 9), and the application deployment view (Figure 10).

For the Smart Charging System, the decomposition view would provide a clear understanding of the various selected components from the reference model and the following modules:

- User Authentication Module: This module manages the authentication and authorization of users, ensuring that only authorized users have access to the charging system;
- User Management Module: This module is responsible for managing the users, including creating and updating user accounts and handling authentication and authorization for accessing the charging stations;

- Charging Session Management Module: This module manages the charging sessions, including starting and stopping the charging process, monitoring the charging progress, and collecting charging data;
- Charging Station Locator Module: Allows users to find available charging stations nearby and reserve a charging slot;
- Payment Processing Module: This module manages the payment transactions for the charging services;
- Energy Management Module: This module monitors and manages the energy usage and distribution for the charging stations;
- Charging Scheduling Module: This module allows users to schedule their charging sessions in advance and receive reminders;
- Charging Station Management Module: This module is responsible for managing the charging stations, including monitoring their status and handling scheduling and billing for the charging sessions;
- Energy Grid Integration Module: This module connects the charging system to the energy grid, enabling the charging stations to access grid power as needed;
- Load Balancing Module: This module balances the load on the charging stations, ensuring that they operate at optimal levels;
- Maintenance Module: This module manages the maintenance tasks for the charging stations, ensuring that they are kept in good working condition;
- Data Analytics Module: This module gathers and analyzes data from the charging system, providing valuable insights into the system's performance and usage. This information could be used to improve the system and better meet users' needs.

Each module is interconnected and interdependent, working together to form a comprehensive smart charging system. The decomposition view helps to clearly illustrate these relationships and the functional responsibilities of each module.

The layered view is then defined based on the selected features and the related modules, as shown in Figure 9.

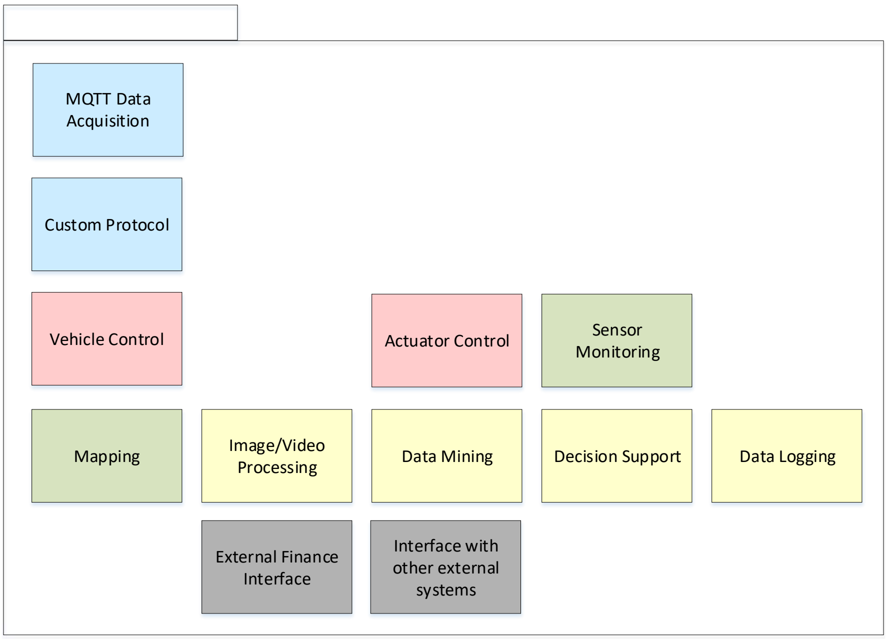

Figure 8. Decomposition View for Smart Charging System.

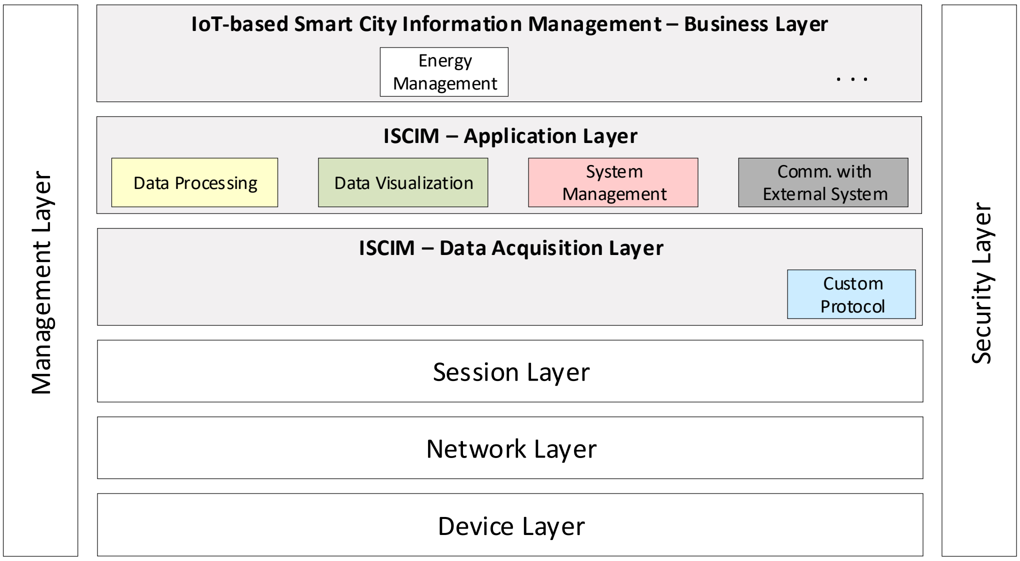

Figure 9. Layering View for Smart Charging System.

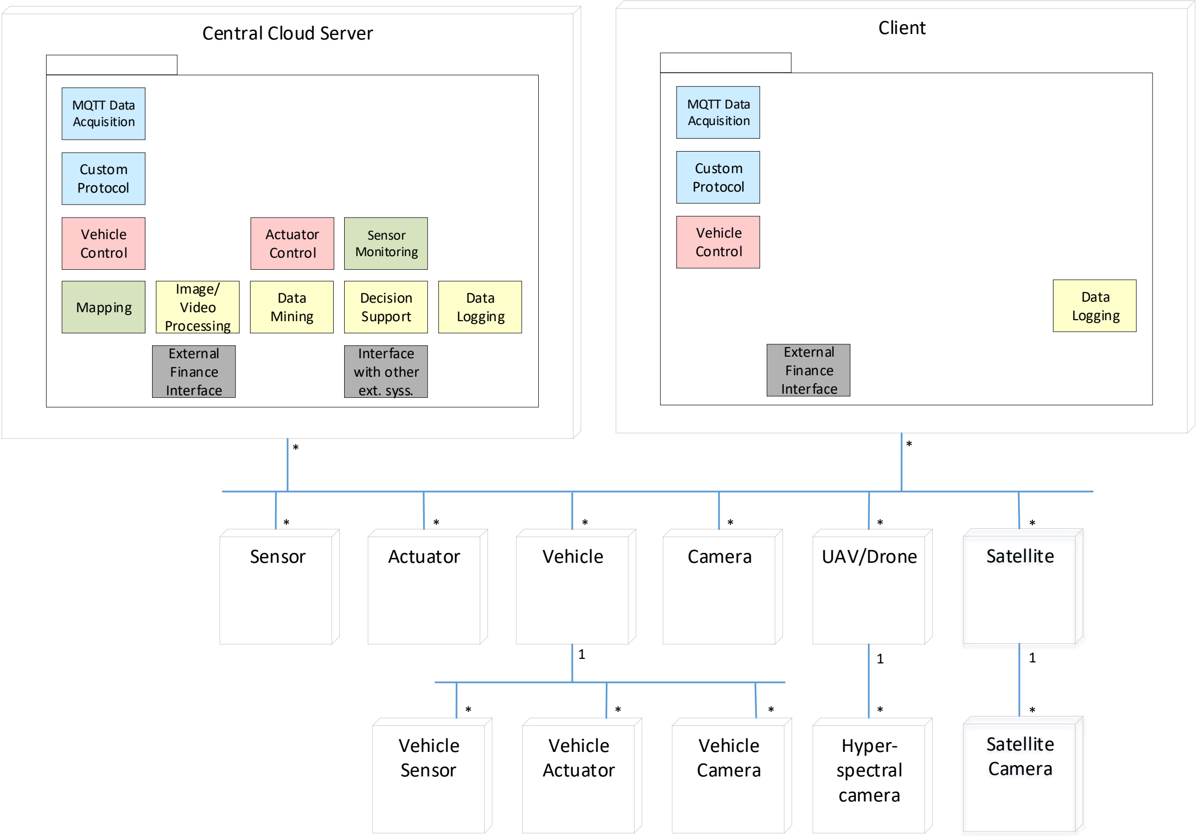

Figure 10. Deployment View for Smart Charging System (1 and * stands for one to many relation).

The deployment view of the smart charging system can be derived from the reference deployment view and is depicted as a network of various devices and components (Figure 10). In this deployment view, the following components can be included:

- General Cloud Server: This component represents the central server that manages the charging system. It collects and processes data from the other components and provides centralized control and management functions;

- Client Node: The client node represents the end-users or consumers who access the charging system. This component may include a mobile application or web interface for customers to monitor and control their charging process;
- Sensor: Sensors collect environmental data, such as charging station availability, energy consumption, and weather conditions;
- Actuator: Actuators control and manipulate the environment, such as turning on and off charging stations, adjusting charging speed, and controlling charging prices;
- Vehicle: This component represents the electric vehicles (EVs) being charged using the system;
- Camera: Cameras can monitor the charging process and detect and prevent unauthorized access to the charging stations;
- UAV/Drone Node: UAVs or drones can collect data from the environment, such as monitoring traffic patterns, collecting air quality data, and monitoring charging stations in remote locations.

These components can be connected through a network, such as a Wi-Fi or cellular network, to allow for communication and data exchange. The deployment view allows system architects and developers to understand the physical distribution and interconnections of the components and to plan for the infrastructure and resources necessary for supporting the smart charging system. The deployed modules to each of the nodes are shown in color to trace these to the modules of the decomposition view. As it can be observed from Figure 10, several modules are deployed on the central cloud server and the client nodes.

6.4. Evaluation

As stated in the case study protocol, we considered the approach's effectiveness and practicality. Evaluating these metrics using the smart charging system case study would provide insights into the viability and potential for real-world implementation of the architecture design method in similar scenarios.

Evaluating the effectiveness of the approach in the case study on the smart charging system involved assessing the extent to which the developed architecture design method meets the desired objectives and solves the problem statement. For real-world projects, this can be completed by comparing the results of the data analysis to the requirements and design documents and verifying the consistency between the desired and actual outcomes. In this paper, we used the requirement as provided in the literature. Furthermore, we designed the architecture based on the provided reference architecture, which was also derived using a domain-driven design approach. It is possible to conclude after evaluating the case study, that the developed architecture design method for the IoT-based smart charging system effectively achieved its goals. Furthermore, the qualitative data analysis results showed a strong agreement among the project managers and system architects that the approach effectively addresses the requirements of the smart charging system.

The practicality of the approach in the case study can be evaluated by considering factors such as the ease of use, scalability, and adaptability of the architecture design method. This can be completed by examining the feedback from the project managers and system architects during the interviews and the Likert scale evaluation and determining the level of agreement with the method's usability and versatility. It is also important to consider the feasibility of the deployment view, including the components' compatibility and the system's overall cost-effectiveness. In our case study, we evaluated the approach ourselves. In terms of practicality, the approach was also deemed to be highly practical. The approach can be evaluated for future projects to consolidate these findings.

7. Discussion

In this section, we will discuss the key findings of our research and their implications for the design of IoT-based smart cities. First, our proposed architecture for IoT-based smart cities is based on a layered structure, with the physical layer consisting of IoT devices, the network layer comprising communication protocols, and the application layer comprising various smart city services. The architecture also includes a management layer for monitoring and controlling the overall system. We have provided detailed descriptions of each layer, including the challenges and potential solutions for their implementation.

One of the key challenges in designing an IoT-based smart city is the integration of various technologies and devices. Our proposed architecture addresses this challenge by providing a standardized and modular approach to integrating IoT devices and services. This allows for the seamless integration of new devices and services as they become available without disrupting the overall system.

Another important aspect of our proposed architecture is the inclusion of a management layer for monitoring and controlling the overall system. This allows for the efficient and effective management of the smart city, including monitoring system performance and identifying and resolving any issues.

There are several potential threats to the validity of this research. One threat is the limited scope of the case study used to illustrate the proposed architecture. While the case study provides a detailed analysis of a smart city, it may not represent all smart cities, and the proposed architecture may not apply to all smart city scenarios. Another threat is the potential for biases in selecting IoT devices and communication protocols. Therefore, further research is needed to evaluate the proposed architecture in different smart city scenarios and with various IoT devices and communication protocols.

Another limitation of this research is that it does not consider the potential ethical and privacy concerns that may arise from using IoT devices in smart cities. The collection and use of data from IoT devices in smart cities may raise privacy concerns for citizens. The proposed architecture should be designed to address these concerns through the use of appropriate security and privacy measures.

In conclusion, our proposed architecture for IoT-based smart cities provides a comprehensive and modular approach to integrating IoT devices and services, allowing for the efficient and effective management of the smart city. It also addresses the key challenges of security and scalability, providing a blueprint for developing smart cities that can benefit citizens and city management.

8. Related Work

In our earlier work [17–19], we discussed the design of a reference architecture for IoT-based smart farming. Although smart farming and smart cities are based on IoT, it is important to analyze the domain thoroughly since the requirements differ for both domains. On the other hand, we could use the reference architecture for IoT systems for both domains.

In recent years, there has been an increasing interest in developing smart cities, which use advanced technology, data analytics, and the Internet of Things (IoT) to optimize the efficiency of city operations and services. As a result, several studies have been conducted to address the challenges and opportunities associated with smart cities.

Several studies have provided secondary studies, including literature reviews on IoT, for example, [12,13,20–22]. In addition, a specific topic on IoT has been security issues discussed in several literature reviews, such as in [23,24].

Some researchers have proposed architectures and frameworks for designing and developing IoT-based smart cities. For example, in [1], the authors propose a framework for integrating IoT devices into smart cities, focusing on using sensors and actuators to collect and process data. Similarly, in [25,26], the authors propose an architecture for integrating IoT devices in smart cities, focusing on cloud computing and big data analytics.

Another important aspect of smart cities is the use of data and analytics to make informed decisions and improve the city's overall performance. Several studies have been conducted in this area, including [27], which focuses on using data analytics to improve the efficiency of transportation systems in smart cities. Similarly, in [28], the authors propose a framework for using data analytics in smart cities, focusing on machine learning and artificial intelligence techniques. Finally, another study has focused on big data analytics for smart cities [29].

In addition to these specific studies, there have also been some general surveys and reviews of the literature on smart cities. For example, in [30], the authors provide an overview of the various challenges and opportunities associated with the development of smart cities, and in [1,2,31], the authors review the literature on the use of IoT in smart cities.

Several studies have focused on specific applications of smart cities using IoT. In [32], the authors discuss the optimization problem of offloading decisions, CPU frequencies, and transmit powers for IoT devices in a multi-mobile edge computing (MEC) server and multi-IoT device cellular network, aimed at minimizing computing pressure on the primary MEC server, energy consumption of the network, and task dropping cost. The problem is formulated as a mixed integer nonlinear program (MINLP) problem, which is difficult to solve due to strongly coupled constraints and discrete integer variables. To solve this problem, a deep reinforcement learning (DRL)-based optimization algorithm is proposed, considering the dynamic nature of the environment. The simulation results show the effectiveness of the proposed algorithm.

In [33] the implementation of a model a smart city with six elements is discussed, namely a smart garbage system, a smart irrigation system, a smart building, a smart parking system, a restaurant menu ordering system, and a manhole detection and monitoring system. The project aims to make advancements in these elements by enabling technologies such as Internet of Things (IoT), embedded systems, and networking. The goal is to make all these systems automatic without any human involvement, thus making life easier. The paper highlights the importance of IoT in combining various sensors with embedded devices to create solutions for real-time problems and connect with the internet society.

Study [34] addresses the gap in the literature that focuses on the four sides of the problem in the automation strategy of smart cities, which includes video surveillance, algorithms, datasets, and embedded systems for edge vision computing. The latest datasets, algorithms, and recent advancements in embedded systems are discussed in the paper, and future trends and challenges in this field are addressed. The paper emphasizes the significance of IoT systems in collecting big data analytics to gain insights for automation strategies in smart cities.

Study [35] proposes an efficient data collection algorithm for an IoT-based smart water meter system, which reduces the number of packet transmissions to save energy. The system components include Arduino Uno, Wi-Fi-ESP8266, and water flow sensors, and the algorithm is implemented and analyzed using MATLAB. Results show that the applied algorithm can save up to 93% of energy consumption in all nodes, making it a promising solution for energy-efficient smart water metering applications.

In reference [35] the authors propose an intelligent proficient data collection approach (IPDCA) for large-scale smart city environments that uses public vehicles as mobile data collectors to collect data from multiple access points and send it back to a central base station. The proposed approach also utilizes a modified Bat algorithm for path finding of data collectors and a multi-objective fitness function to select them in smart city settings. The effectiveness of the proposed approach is demonstrated through simulations.

Several European research projects on IoT provided facilities that can be exploited for the development of interoperable smart city applications. For example, SymBiote is an EU-funded project that aims to develop an open and secure middleware platform for the Internet of Things (IoT) [36]. The project seeks to address the challenges of heterogeneity, scalability, and security that arise in IoT systems. The SymBiote middleware platform is designed to enable seamless communication between different IoT devices and systems, regardless of their manufacturer or technology used. It provides a set of services and tools for managing data, security, and communication between devices and systems. One of the key objectives of the SymBiote project is to promote interoperability and standardization in IoT systems. Similarly, Open IoT is building a novel open source platform for the Internet of

Things (IoT), which includes a middleware infrastructure to support flexible configuration and deployment of interoperable IoT systems [37].

Snap4City is a platform for building and deploying smart city applications, which aims to provide a user-friendly and customizable environment for developing IoT applications. The platform includes a range of tools and services for collecting, processing, and analyzing data from various sources, including IoT devices, social media, and open data sources. One of the key features of the Snap4City platform is its support for interoperability, and likewise provides support for a wide range of protocols and standards, including MQTT, CoAP, and HTTP, making it compatible with a wide range of IoT devices and systems.

In general, the related work in smart cities has focused on a wide range of topics, including IoT, data analytics, and cloud computing in smart cities. In addition, the proposed architectures and frameworks have focused on different aspects of smart cities, including transportation systems, energy systems, public safety, healthcare, and waste management. However, there is still a need for more research on the design and development of IoT-based smart cities, particularly in integrating various technologies, communication protocols, and data processing and management systems.

9. Conclusions

This paper proposed a novel systematic method for guiding the system architecture design of IoT-based smart cities. First, we adopted a feature-driven domain analysis to model the smart city requirements, followed by the design of a reference architecture for IoT-based smart cities. The proposed architecture includes a layered structure, with the physical layer consisting of IoT devices, the network layer comprising communication protocols, and the application layer comprising various smart city services. The architecture also includes a management layer for monitoring and controlling the overall system.

We provided detailed descriptions of each layer, including the challenges and potential solutions for their implementation. As a result, the proposed architecture can serve as a blueprint for developing smart cities, enabling the integration of various IoT devices and services to benefit citizens and city management. Furthermore, it can be modified and enhanced to meet the specific needs of different cities and communities.

The paper also contributes to the existing literature on smart cities by presenting a comprehensive overview of the key elements, including the internet of things and the challenges of designing smart cities. It also compares the proposed architecture with existing solutions and discusses future research directions. Overall, this paper provides a valuable contribution to the field of smart city design and implementation.

This paper provides a novel contribution to the field of smart city development, and we believe it will be useful for researchers, practitioners, and policymakers alike. Future work could include the implementation and testing of the proposed architecture in real-world scenarios, as well as the exploration of other potential use cases for IoT-based smart cities.

Author Contributions: Conceptualization, B.T., Ö.K. and T.Ç.; methodology, B.T., Ö.K. and T.Ç.; validation, B.T., Ö.K. and T.Ç.; investigation, B.T., Ö.K. and T.Ç.; writing—original draft preparation, B.T., Ö.K. and T.Ç.; writing—review and editing, B.T., Ö.K. and T.Ç. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Glossary

Architecture framework	A set of principles, standards, and guidelines for designing and developing architecture, which provides a common language and structure for communication and collaboration among stakeholders.
Big data	A term used to describe the large volume, variety, and velocity of data that is generated and collected from various sources, including IoT devices, social media, and other digital systems.
Big data analytics	The process of examining large and complex data sets to uncover hidden patterns, correlations, and other insights.
Cloud computing	A model of computing that involves delivering on-demand computing services over the internet, such as storage, processing, and applications, on a pay-per-use basis.
Communication protocol	A set of rules and standards that govern the exchange of data between two or more devices or systems.
Data analytics	The process of collecting, processing, and analyzing large amounts of data to identify patterns, correlations, and insights that can be used to improve decision-making and performance.
Data processing	The manipulation and analysis of data to extract meaningful insights or information.
Domain analysis	A systematic study of the characteristics and requirements of a particular domain, such as smart cities or IoT, to identify common patterns and features.
Internet of Things (IoT)	A network of physical objects or devices, such as sensors and actuators, that are connected to the internet and can exchange data with other devices and systems.
Interoperability:	The ability of different systems or components to communicate, exchange data, and work together seamlessly.
Machine learning	A subset of artificial intelligence (AI) that involves training computer systems to learn from data and improve their performance on a specific task over time.
Reference architecture	A blueprint or framework for designing and developing a system or application which specifies the components, interfaces, and standards to be used.
Sensor network	A collection of sensors that are connected to each other and to a central system which can be used to collect and process data from the physical environment.
Smart city	A city that uses advanced technology and data analytics to optimize the efficiency of city operations and services, such as transportation, energy, public safety, and waste management.
Systemrequirements	The functional and non-functional needs of a system, including its performance, reliability, security, and scalability.

References

1. Lee, Y.; Pervin, F. Smart cities: A literature review. *J. Urban Technol.* **2018**, *25*, 1–19. [CrossRef]
2. Zanella, A.; Bui, N.; Castellani, A.; Vangelista, L.; Zorzi, M. Internet of things for smart cities. *IEEE Internet Things J.* **2014**, *1*, 22–32. [CrossRef]
3. Guo, Y.; Li, Y.; Li, X. Smart city architecture: A review. *IEEE Access* **2020**, *8*, 166879–166899. [CrossRef]
4. Liu, X.; Zhang, Y.; Sun, H.; Wang, X. A survey on Internet of Things for smart cities. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 809–828.
5. Ahmad, K.; Maabreh, M.; Ghaly, M.; Khan, K.; Qadir, J.; Al-Fuqaha, A. Developing future human-centered smart cities: Critical analysis of smart city security, Data management, and Ethical challenges. *Comput. Sci. Rev.* **2022**, *43*, 100452. [CrossRef]
6. Kamal, K.; Al-Fuqaha, A.; Guizani, M.; Mohammadian, M.; Aledhari, M. Internet of things for smart cities. *IEEE Commun. Mag.* **2015**, *53*, 40–47.
7. Silva, J.R.; Couto, J.A.F. Smart cities: Vision, state of the art and future challenges. *IEEE Commun. Mag.* **2011**, *49*, 34–40.
8. Al-Turjman, F.; Zahmatkesh, H.; Shahroze, R. An overview of security and privacy in smart cities' IoT communications. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e3677. [CrossRef]
9. ITU. The internet of things. *ITU Internet Rep.* **2005**, *2005*, 212.

10. Tekinerdogan, B. Obstacles of System-of-Systems. In Proceedings of the 2022 IEEE International Symposium on Systems Engineering (ISSE), Vienna, Austria, 24–26 October 2022; pp. 1–7. [CrossRef]
11. Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2347–2376. [CrossRef]
12. Pandya, H.B.; Champaneria, T.A. Internet of things: Survey and case studies. In Proceedings of the 2015 International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO), Visakhapatnam, India, 24–25 January 2015; pp. 1–6. [CrossRef]
13. McEwen, A.; Cassimally, H. *Designing the Internet of Things*; Wiley: Hoboken, NJ, USA, 2014.
14. Clements, P.; Bachmann, F.; Bass, L.; Garlan, D.; Ivers, J.; Little, R.; Merson, P.; Nord, R.; Stafford, J. *Documenting Software Architectures*, 2nd ed.; Adddison-Wesley: New York, NY, USA, 2010.
15. Runeson, P.; Höst, M. Guidelines for conducting and reporting case study research in software engineering. *Empir. Softw. Eng.* **2008**, *14*, 131–164. [CrossRef]
16. Köksal, Ö.; Tekinerdogan, B. Architecture design approach for IoT-based farm management information systems. *Precis. Agric.* **2019**, *20*, 926–958. Available online: https://doi-org.ezproxy.library.wur.nl/10.1007/s11119-018-09624-8 (accessed on 21 March 2023). [CrossRef]
17. Köksal, Ö.; Tekinerdogan, B. Feature-driven domain analysis of session layer protocols of internet of things. In Proceedings of the 2nd International Congress on Internet of Things, ICIOT 2017, Honolulu, HI, USA, 25–30 June 2017; pp. 105–112.
18. Tekinerdogan, B.; Köksal, Ö. Pattern Based Integration of Internet of Things Systems. In *Internet of Things—ICIOT 2018*; Springer: Cham, Switzerland, 2018; pp. 19–33.
19. Al-Fuqaha, A.; Guizani, M.; Mohammady, M.; Aledhari, M.; Ayyash, M. Internet of Things: A survey of topics, technologies, and applications. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2323–2360. [CrossRef]
20. Atzori, L.; Iera, A.; Morabito, G. The Internet of Things: A survey. *Comput. Netw.* **2010**, *54*, 2787–2805. [CrossRef]
21. Ahmed, N.; Al-Fuqaha, A. Internet of Things: A survey on enablers, challenges, and open research topics. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2347–2376. [CrossRef]
22. Kumar, G.S.S.R.; Thai, D.T.D.; Buyya, R. A survey of Internet of Things (IoT) security research. *J. Netw. Comput. Appl.* **2011**, *34*, 1–19.
23. Al-Sultan, K.; Al-Dubai, A. A survey of internet of things security challenges and solutions. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1517–1542. [CrossRef]
24. Johnson, A. An architecture for the integration of IoT devices in smart cities: Emphasizing the use of cloud computing and big data analytics. *J. Smart City Dev.* **2022**, *5*, 45–54.
25. Smith, J. Integrating IoT devices in smart cities: A framework for sensor and actuator-based data collection and processing. *J. Smart City Res.* **2021**, *3*, 12–23.
26. Brown, K. Data analytics for improving transportation systems in smart cities. In Proceedings of the International Conference on Smart City Technologies, Virtual Conference, 28 September–1 October 2020; pp. 78–85.
27. Wilson, D. A framework for the use of data analytics in smart cities: Focusing on machine learning and artificial intelligence techniques. *J. Smart City Appl.* **2021**, *7*, 123–135.
28. Zhang, Y.; Liu, X.; Wang, X. Smart city big data: A survey. *IEEE Access* **2019**, *7*, 83754–83771. [CrossRef]
29. Park, S.; Kim, Y. The challenges and opportunities of smart cities: A comprehensive review. *J. Smart City Plan.* **2019**, *8*, 1–12.
30. Lee, J.; Chen, L. The use of IoT in smart cities: A literature review. In Proceedings of the International Conference on Internet of Things and Smart Cities, Virtual Conference, 28 September–1 October 2020; pp. 56–63.
31. Tian, K.; Chai, H.; Liu, Y.; Liu, B. Edge Intelligence Empowered Dynamic Offloading and Resource Management of MEC for Smart City Internet of Things. *Electronics* **2022**, *11*, 879. [CrossRef]
32. Avinash, P.; Vamsi, B.K.; Srilakshmi, T.; Kishore, P.V.V. Smart Control System for Smart City using IoT. *Int. J. Adv. Comput. Sci. Appl.* **2021**, *12*, 408–414. [CrossRef]
33. Ezzat, M.A.; Abd El Ghany, M.A.; Almotairi, S.; Salem, M.A.-M. Horizontal Review on Video Surveillance for Smart Cities: Edge Devices, Applications, Datasets, and Future Trends. *Sensors* **2021**, *21*, 3222. [CrossRef] [PubMed]
34. Al-Madhrahi, R.; Abdullah, J.; Alduais, N.A.M.; Mahdin, H.B.; Saad, A.M.H.Y.; Nasser, A.B.; Alduais, H.S. An Efficient IoT-based Smart Water Meter System of Smart City Environment. *Int. J. Adv. Comput. Sci. Appl.* **2021**, *12*, 420–428. [CrossRef]
35. Osamy, W.; Khedr, A.M.; El-Sawy, A.A.; Salim, A.; Vijayan, D. IPDCA: Intelligent Proficient Data Collection Approach for IoT-Enabled Wireless Sensor Networks in Smart Environments. *Electronics* **2021**, *10*, 997. [CrossRef]
36. SymBiote. (n.d.). SymBiote Project Website. 2023. Available online: https://www.symbiote-h2020.eu/ (accessed on 18 March 2023).
37. OpenIoT. GitHub Repository. Available online: https://github.com/OpenIotOrg/openiot/ (accessed on 18 March 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.