uc3m | Universidad Carlos III de Madrid

University Degree in Data Science and Engineering and Telecommunication Technologies Engineering
2024 – 2025

Bachelor thesis

Building a Scalable and Decentralized Parking Management Solution for Multi-Community Deployment

---

Andrés Navarro Pedregal

Tutor
David Larrabeiti López

Leganés, 2025

ABSTRACT

this is an abstract

Keywords: keyword1, keyword2, keyword3

ACKNOWLEDGMENTS

Thanks

AGRADECIMIENTOS

Gracias

TABLE OF CONTENTS

I Introduction 1

1 Motivation 2

2 Statement of the problem 3

3 Objectives 4

4 Document Structure 5

5 Methodological Framework 6

II Theoretical Background 9

6 Cloud Computing 10
6.1 Definition and Key Concepts . . . 10
6.2 Advantages of Cloud Computing . . . 10
6.3 Types of Cloud Computing . . . 11
6.3.1 Deployment Models . . . 11
6.3.2 Service Models . . . 11

7 Internet of Things 12
7.1 Architecture of IoT Systems . . . 12
7.2 Benefits of IoT . . . 12
7.3 Challenges and Limitations . . . 12

III State of the art 14

8 Historical Development 15

9 Types & Technologies 16

10 Modern Trends 17

IV Methodology 19

11 Requirements 20
11.1 User Requirements . . . 20
11.2 System Requirements . . . 21

12 Design 22
12.1 Terminals . . . 22
12.1.1 Cameras . . . 22

12.1.2 Computer 22
12.1.3 Networking 23
12.1.4 Rele Extension board 23
12.2 Server 23
12.3 Monitoring 23
12.4 Deployment 23

13 Implementation **24**

14 Testing **25**

V Results **27**

VI Conclusions **29**

15 Conclusions **30**

16 Future work **31**

17 Socio-economic environment **32**

18 Regulatory Framework **33**
18.1 Applicable Legislation 33
18.1.1 General Data Protection Regulation (GDPR) 33
18.1.2 Data Retention Policies 33
18.1.3 ePrivacy Directive 33
18.1.4 Directive on Security of Network and Information Systems (NIS 2 Directive) 34
18.1.5 Artificial Intelligence Act 34
18.2 Regulatory Challenges and Implications 34
18.3 Conclusion 34

LIST OF FIGURES

5.1 Methodological framework based on the V-model from INCOSE with the different stages of the project development process[12] 7

LIST OF TABLES

LIST OF ACROYNMS

AI	Artificial Intelligence.
AWS	Amazon Web Service.
GDPR	General Data Protection Regulation.
INCOSE	International Council on Systems Engineering.
IoT	Internet of Things.
ML	Machine Learning.
NIS	Network and Information Systems.
PMS	Parking Management System.
PoE	Power over Ethernet.

Part I

Introduction

1. MOTIVATION

The rapid growth of urban populations and the corresponding increase in vehicle ownership have significantly intensified the challenges associated with parking management in cities worldwide. For instance, the dramatic rise in car usage in Madrid has substantially contributed to heightened levels of environmental pollution [1]. Despite the number of parking spaces in Spain remaining stable between 2014 and 2020 [2], the escalating demand for parking has resulted in higher costs, prolonged search times, increased traffic congestion, and elevated urban pollution levels.

The inadequacies of traditional, manual Parking Management Systems (PMSs) have further highlighted the need for innovative solutions in this domain. Studies highlight the importance of user-friendly, secure, and reliable systems, with these factors playing a crucial role in influencing parking behavior and preferences [3]. Consequently, the development of effective PMSs remains an essential area of research, particularly as urbanization continues to accelerate.

Advancements in technology provide new opportunities to address these challenges. The proliferation of internet-connected devices, growing by 20% annually [4], has catalyzed the emergence of Internet of Things (IoT)-based PMSs. These systems aim to mitigate traffic congestion and reduce urban pollution while promoting sustainable and efficient urban mobility. By leveraging real-time data and automation, modern PMSs offer transformative potential for urban planning and management.

Integrating these advanced systems into urban environments offers multiple benefits, including decreased traffic congestion, reduced pollution, enhanced security, and an improved quality of life for residents. Additionally, PMSs enable automated parking space management, making them adaptable for a wide range of applications, including urban centers, residential communities, and commercial buildings.

Despite these advancements, the widespread adoption of PMSs in Spain has faced significant obstacles. Many existing systems rely heavily on human intervention, leading to delays, insufficient information dissemination, and limited control over parking space availability. This dependence on manual operations not only impairs system efficiency but also increases operational costs.

Emerging technologies have inspired the development of innovative PMSs, such as RFID-based systems [5], IoT-enabled solutions [6], and intelligent parking systems employing image processing techniques [7]. However, these systems frequently rely on centralized architectures, which are prone to scalability and reliability issues, particularly during service disruptions. Moreover, many current solutions lack the flexibility needed to accommodate the specific requirements of diverse user groups, limiting their broader applicability.

Addressing these limitations, the primary objective of this project is to design and implement a fully distributed and autonomous PMS that overcomes the challenges inherent in existing systems. This new approach emphasizes scalability, reliability, and adaptability, with a particular focus on meeting the demands of next-generation smart cities. Through this work, the project aims to contribute to the development of sustainable, efficient, and user-centric parking solutions for urban environments.

2. STATEMENT OF THE PROBLEM

The lack of efficient PMSs is a pervasive problem in modern cities, exacerbated by the growing number of vehicles and the resulting increase in urban population. Traditional manual PMSs have proven inadequate in addressing the complexities of modern urban parking, leading to higher costs, longer search times, traffic congestion, and elevated levels of urban pollution. The ongoing development of PMSs remains a crucial research area, as citizens favor systems that prioritize user-friendliness, security, and reliability [3] [8] [9] [10].

This project aims to address the gap by designing and implementing a fully distributed PMS tailored for the next generation of smart cities. The proposed system will focus on addressing the inefficiencies and challenges inherent in current PMSs through a distributed approach that leverages modern technologies.

Given the broad nature of this problem, the scope of this project will be narrowed to a specific environment and use case, as outlined below:

- **Environment:** The system will be designed for urban environments with high vehicle density and limited parking spaces. The case study will focus on a city center with a high volume of vehicles and limited parking availability, reflecting the challenges faced by many modern cities.
- **Atmospheric Conditions:** The system will be designed to operate under typical urban atmospheric conditions, with no specific constraints on temperature, pressure, or humidity. The system will be designed to function in a variety of weather conditions, including rain, heat, and cold.
- **Operational Parameters:** The system will be designed to manage parking spaces in a city center, with a focus on optimizing space utilization, reducing search times, and minimizing traffic congestion. The system will be scalable to accommodate a large number of parking spaces and users.
- **Hardware:** The system will be designed to leverage modern technologies such as IoT devices, sensors, and cloud computing to provide real-time monitoring and management of parking spaces. The system will be designed to be cost-effective, scalable, and user-friendly.

3. OBJECTIVES

The primary objective of this bachelor thesis is to design and implement a fully distributed parking management system tailored for the next generation of smart cities. This project aims to address the inefficiencies and challenges inherent in current parking management systems through a distributed approach that leverages modern technologies. The objective of this project can be broken down into the following specific goals:

- Conduct a comprehensive study of existing parking management systems, identifying their main problems and inefficiencies. This involves understanding user requirements, analyzing the technologies employed, and evaluating system effectiveness.
- Analyze the technologies currently used in parking management systems to determine their suitability for a distributed architecture. This includes examining sensors, IoT devices, communication protocols, and data processing methods to identify the most suitable technologies for the proposed system.
- Design the overall infrastructure of the distributed parking management system. This encompasses defining the system architecture, selecting appropriate technologies, and developing detailed design specifications to ensure a scalable, secure, and user-friendly system.
- Develop the system by adhering to a structured methodology that includes phases of planning, design, implementation, testing, deployment, and maintenance. Each phase will follow best practices to ensure the system's robustness, efficiency, and reliability.
- Evaluate the implemented system based on various criteria such as performance, scalability, security, usability, reliability, availability, and cost. This comprehensive analysis will help in assessing the effectiveness of the system and identifying areas for improvement.

  and test the proposed system in a real-world scenario, such as a city center with high vehicle density and limited parking spaces, to validate its effectiveness and identify potential issues.

By achieving these objectives, the thesis aims to contribute to the advancement of smart city technologies by providing a scalable, secure, and user-friendly parking management solution. The distributed nature of the proposed system is expected to enhance its performance and reliability, making it a viable option for modern urban environments.

4. DOCUMENT STRUCTURE

TODO: Write this chapter when all the document is finished.

5. METHODOLOGICAL FRAMEWORK

To address the research objectives outlined in this thesis, a structured methodological framework is adopted to ensure a comprehensive and rigorous development process. This chapter presents the methodological approach employed in this research, which is grounded in the V-model as established by the International Council on Systems Engineering (INCOSE) [11]. The V-model provides a robust framework for project development, facilitating timely and budget-compliant completion through a comprehensive development process.

The methodological framework is composed of seven key stages, each addressing a distinct aspect of the project development process. The stages are designed to ensure clear validation and verification of initial requirements at every stage, guaranteeing that the proposed solution aligns with the project's objectives.

1. **User Requirement Identification**: This stage involves a detailed analysis of the problem statement to identify primary issues and potential solutions. User requirements are defined to ensure that the proposed solution aligns with the project's objectives. This stage is crucial in establishing a clear understanding of the project's scope and requirements.
2. **System Design and Architecture**: Based on the identified user requirements, the system architecture is developed, ensuring that the proposed solution is feasible and aligns with the project's objectives. This stage involves a high-level overview of the system components and their interconnections. Requirements are formulated to satisfy the previously defined solution requirements.
3. **Component Design and Development**: Building upon the high-level architecture of the solution, a more detailed approach is outlined for each component, taking into account their specific power and data transmission needs. This stage culminates in a comprehensive architecture of the solution, providing a detailed overview of the components, their selection rationale, and their interconnections.
4. **Implementation and Manufacturing**: The proposed solution is implemented and manufactured using available tools, integrating the necessary electrical components. This stage includes a detailed description of the implementation process, including the tools and materials used, as well as the integration of electrical components. The development of software and hardware components is also detailed.
5. **Component Testing and Verification**: The functionality of each component is verified in a standalone mode, with detailed information provided regarding the verification process. This stage ensures that each component functions as intended, paving the way for system integration.
6. **System Testing and Integration**: The methodology for conducting system tests and subsequent analyses is elaborated. System integration is performed by assessing communication between module pairs to ensure that data can be transmitted freely and utilized effectively.
7. **Acceptance Testing and Validation**: Validation of the initial requirements is conducted to confirm that all solution requirements have been met. This stage also in-

cludes preparations for potential future enhancements, ensuring the solution's scalability and adaptability.

A graphical representation of the V-model is provided in Figure 5.1 to illustrate the methodology's structure and relationship between the various stages.

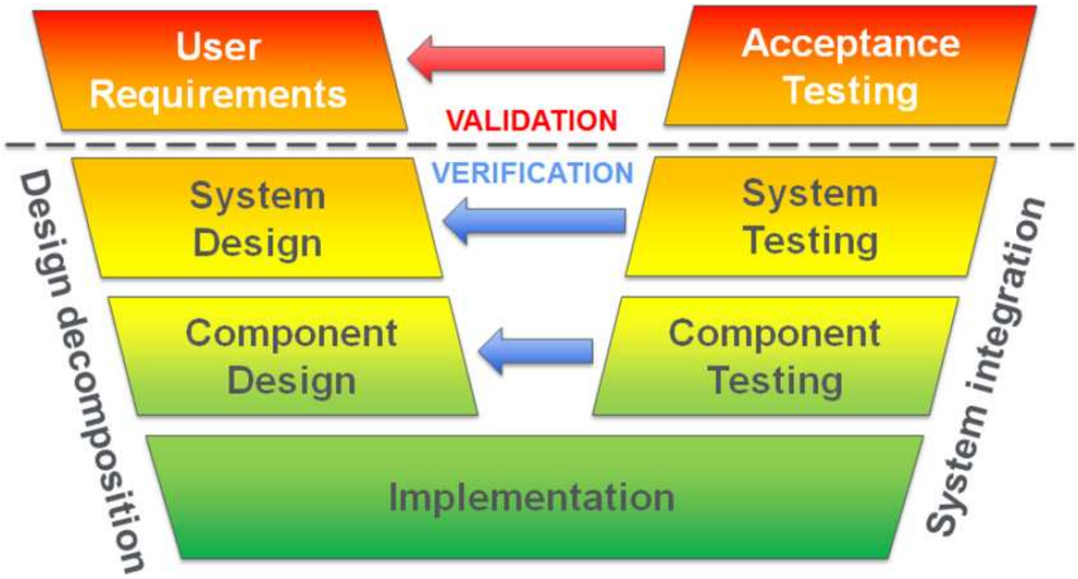

Figure 5.1. *Methodological framework based on the V-model from INCOSE with the different stages of the project development process[12]*

The V-model provides a structured approach to project development, ensuring that all aspects of the project are considered and addressed. By following this methodological framework, this research aims to develop a comprehensive and effective solution for the design and implementation of a fully distributed parking management system tailored for the next generation of smart cities.

Part II

Theoretical Background

6. CLOUD COMPUTING

Cloud computing is a technological paradigm that enables the delivery of computing services, including servers, storage, databases, networking, software, and analytics, over the internet. This model shifts the management of physical infrastructure and resources from on-premises systems to remote data centers managed by cloud service providers. Cloud computing is fundamental to modern digital ecosystems, supporting a wide range of applications from personal use to large-scale enterprise operations.

6.1 Definition and Key Concepts

Cloud computing leverages a network of remote servers hosted on the internet to store, manage, and process data, rather than relying on local computers or private data centers. It offers resources on-demand, enabling users to scale operations according to their needs. The essential characteristics of cloud computing include:

- **On-Demand Self-Service:** Users can access computing resources as needed without requiring human interaction with service providers.
- **Broad Network Access:** Services are accessible over a network, typically the internet, using various devices such as smartphones, tablets, and laptops.
- **Resource Pooling:** Resources are pooled to serve multiple users, ensuring efficiency and scalability.
- **Rapid Elasticity:** Resources can be elastically provisioned and released, enabling scalability according to demand.
- **Measured Service:** Cloud systems automatically control and optimize resource use, charging users based on their consumption.

6.2 Advantages of Cloud Computing

The adoption of cloud computing offers several advantages that have contributed to its widespread popularity. One of the most significant benefits is its scalability and flexibility. Cloud computing enables organizations to adjust their resource usage dynamically based on real-time needs, eliminating the need for over-provisioning and supporting workloads of varying demands effectively. This adaptability ensures that resources are utilized efficiently, minimizing waste and optimizing performance.

Cost efficiency is another critical advantage. By using cloud services, organizations can avoid the significant capital expenditures associated with purchasing and maintaining hardware and software. Instead, they incur operational costs only for the resources they consume, making cloud computing an economically viable solution for businesses of all sizes. Additionally, cloud computing enhances accessibility and collaboration. Since cloud-based applications and services are accessible from anywhere with an internet connection, distributed teams can collaborate seamlessly, a feature that has become indispensable in today's increasingly globalized and remote work environments.

Reliability is also a key strength of cloud computing. Leading cloud providers ensure high levels of availability by deploying redundant systems across geographically dispersed data centers, thereby minimizing downtime and enhancing disaster recovery capabilities. Furthermore, security is a paramount concern addressed by cloud providers. They invest significantly in advanced measures such as encryption, firewalls, and regular security audits, ensuring robust protection against cyber threats and unauthorized access. These combined advantages make cloud computing a transformative technology, supporting innovation and operational efficiency in diverse sectors.

6.3 Types of Cloud Computing

Cloud computing services are categorized into different types based on their deployment models and the services they provide. These distinctions allow organizations to select a cloud strategy that aligns with their specific requirements.

6.3.1 Deployment Models

- **Public Cloud:** A public cloud is owned and operated by third-party providers, delivering resources over the internet. Examples include Amazon Web Service (AWS), Microsoft Azure, and Google Cloud Platform. It offers cost-effectiveness and scalability but may pose data sovereignty concerns for certain organizations.
- **Private Cloud:** A private cloud is used exclusively by a single organization. It offers greater control over data and infrastructure, making it suitable for industries with strict regulatory requirements.
- **Hybrid Cloud:** A hybrid cloud combines public and private cloud environments, enabling organizations to benefit from both scalability and control. This model is particularly useful for balancing workloads and maintaining sensitive data on-premises while leveraging the public cloud for other operations.
- **Multi-Cloud:** A multi-cloud strategy involves using multiple cloud providers to mitigate risks associated with vendor lock-in and enhance reliability and performance.

6.3.2 Service Models

Cloud computing services are typically offered in the following models:

- **Infrastructure as a Service (IaaS):** Provides virtualized computing resources over the internet, including servers, storage, and networking. Users manage operating systems and applications while the provider handles the infrastructure.
- **Platform as a Service (PaaS):** Offers a platform that includes hardware, software, and development tools to build, test, and deploy applications. It abstracts the complexities of infrastructure management, enabling developers to focus on coding.
- **Software as a Service (SaaS):** Delivers software applications over the internet on a subscription basis. Users can access the software through a browser without worrying about installation or maintenance.

7. INTERNET OF THINGS

The IoT is a paradigm that establishes a network of interconnected devices equipped with sensors, software, and communication technologies. This enables seamless interaction between the physical and digital realms, facilitating the collection, transmission, and processing of data. Through its ability to deliver enhanced functionality and automation, the IoT has numerous applications, particularly in smart city infrastructure. These applications span healthcare, transportation, energy management, and urban planning.

The main point of IoT is its capacity for interconnectivity, enabling devices to autonomously communicate and share data. Advanced sensing technologies augment this connectivity, capturing real-world metrics such as temperature, motion, or occupancy. The processing of the data is achieved through edge computing, which provides localized analysis, or cloud computing for centralized processing, tailored to meet latency and scalability demands. This processed data is subsequently integrated with actuators, enabling automation of tasks such as the management of parking space access without requiring human intervention.

7.1 Architecture of IoT Systems

The architecture of IoT systems is structured across four principal layers. The perception layer constitutes sensors and actuators, which interface directly with the physical environment to gather data and execute specific actions. Data transmission occurs within the network layer, leveraging communication protocols such as , , or cellular networks. The processing layer transforms raw data into actionable insights through edge or cloud computing methodologies. Finally, the application layer provides interfaces for end-users, ensuring an intuitive and accessible experience tailored to diverse functionalities.

7.2 Benefits of IoT

The adoption of IoT technologies confers significant advantages across multiple sectors, particularly in the realm of parking management systems. Real-time monitoring capabilities, enabled by sensors, deliver continuous updates on parking space availability, reducing search times and alleviating urban congestion. Automation, facilitated by actuators, streamlines operations such as access control and reservation management, minimizing human intervention. Moreover, IoT-driven data analytics empower urban planners with insights to optimize resource allocation and develop informed policies. The energy efficiency inherent in smart systems aligns with sustainability objectives by curbing unnecessary resource consumption.

7.3 Challenges and Limitations

The deployment of IoT systems presents several challenges. Scalability remains a critical concern, particularly in large-scale implementations like city-wide parking systems. The management of sensitive information collected by IoT devices necessitates data security and privacy measures to itigate risks. Interoperability challenges, stemming from disparate manufacturer standards and communication protocols, complicate system integration.

Part III

State of the art

8. HISTORICAL DEVELOPMENT

TODO: Write this chapter

9. TYPES & TECHNOLOGIES

TODO: Write this chapter

10. MODERN TRENDS

TODO: write this chapter

Part IV

Methodology

11. REQUIREMENTS

TODO: Write this chapter

The parking management system developed for this project was designed to ensure that it met the needs of drivers, parking facility managers, and city administrators. The system was designed to address the limitations of current parking management systems, focusing on enhancing scalability, reliability, and user adaptability.

For this purpose, different user requirements were identified, and system requirements were defined to meet these needs. Furthermore, the system was designed to incorporate specific technical and functional specifications to ensure its effectiveness and efficiency.

11.1 User Requirements

The user requirements for the distributed parking management system were identified through a comprehensive analysis of the needs and preferences of drivers, parking facility managers, city administrators and users with disabilities. These requirements were essential to ensure that the system was user-friendly, efficient, and aligned with the objectives of smart city initiatives [13].

The requirements of the primary users were as follows:

For drivers, the system needed to provide real-time information about available parking spaces to minimize search time. This was crucial to reduce traffic congestion and pollution in urban areas. Moreover, drivers expected an easy-to-use interface for quick navigation and more importantly, automatic functionality of the system without the need for human intervention. That way, they did not have to worry about the availability of parking spaces and could focus on other tasks.

Parking facility managers required tools to monitor and manage their facilities efficiently, as well as access to detailed reports and analytics on parking usage patterns to optimize space utilization. A notification system for intrusions such as unauthorized parking or security breaches was also essential to ensure the safety of the parking facilities. Enhanced security measures, including surveillance and access control, were a must-have for them.

On the other hand, city administrators needed a system that could provide insights into parking demand and usage trends to inform urban planning decisions. The system should support the integration of parking data with other smart city initiatives to enhance overall urban mobility and sustainability. Moreover, city administrators required tools to monitor and minimize the environmental impact of parking facilities, such as reducing emissions and energy consumption. The system should also comply with local regulations and standards for data privacy and security such as GDPR [14].

Finally, users with disabilities needed accessibility features such as voice commands, screen readers, and other assistive technologies to ensure that they could use the system effectively. These features were essential to promote inclusivity and ensure that all users could benefit from the distributed parking management system. Moreover, the system should be designed to accommodate users with different needs and preferences, ensuring a seamless user experience for everyone.

11.2 System Requirements

For the distributed parking management system to meet the user requirements, it had to incorporate specific system requirements. These requirements were defined to ensure that the system was scalable, reliable, secure, and user-friendly, aligning with the objectives of smart city initiatives as well as the needs of drivers, parking facility managers, and city administrators. The system requirements were essential to guide the design and development of the distributed parking management system, ensuring that it met the expectations of all stakeholders and delivered the desired outcomes.

The main system requirements were as follows:

For the real-time monitoring of parking space availability, the system needed to incorporate sensors and IoT devices to detect vehicle presence and occupancy. These sensors had to be accurate, reliable, and cost-effective to provide up-to-date information on parking availability.

Reservation was another key requirement, allowing drivers to be assigned a specific number of parking spaces in advance to ensure that they could secure a spot when needed. The reservation system had to be integrated with the monitoring system to ensure that drivers could find available spaces.

The system also needed to be self-sufficient, with automated gate and barrier control to regulate access to parking facilities. It had to be capable of managing multiple parking facilities simultaneously, ensuring that drivers could access the system from different locations. And it had to be able to function without human intervention, reducing the need for manual oversight.

Moreover, the system had to be scalable and flexible, with a distributed architecture that could handle large volumes of data and a growing number of users. It had to support additional number of communities and parking facilities, ensuring that it could adapt to changing requirements. For the system to be effective, modularity was essential, allowing for easy updates and integration of new features as needed.

Security and availability were critical requirements for the system, ensuring that the system was robust and resilient against cyber threats and service interruptions. Interoperability was crucial to ensure compatibility with existing and future smart city infrastructure.

Finally, the user interface had to be intuitive and user-friendly, with mobile accessibility for drivers and web-based interfaces for facility managers and city administrators. And accessibility features were needed to support users with disabilities, ensuring that the system was inclusive and accessible to all.

12. DESIGN

This chapter outlines the architecture of the system design, detailing the different subsystems and components to meet the requirements detailed in Chapter 11. The system designed is composed of two main components, a server, in charge of manageming the users, authenticating them, and relaying the information form the users, and the terminals, which are the brains of the system and hold the main control for each community. It is worth to note that while the terminal is deployed in every community, the server is only deployed once and can be used for every user in the system. The architecture of the system can be seen in **TODO:** add image with architecture. Note that there are other complementary elementes that they will be later explained.

TODO: add image of architecture and explain

12.1 Terminals

The terminals are the core part of the deployment and is where all the processing is done to manage each community. In order to provide the necesary capabilities for the community such as opening the garage doors and detecting the cars the following components are needed.

12.1.1 Cameras

Two cameras per door to be able to detect the cars when they are close to the garage door. This cameras are the ones in charge of sending live video to be able to detect the cars and later identify the licence plates. The requirements for the cameras are that they are wether proof and can withstand heavy rain, can send live feed and be connected to a central computer, and have support for infrared detection for low light events such as inside garage parkings or night. Different models of cameras where studied but the final camera used for this project was the **TODO:** add camera because of **TODO:** add reason

TODO: add image of camera

Moreover, even thought the camera has infrared capabilities, the infrared light of the camera is pretty weak. Therefore an anditional infrared led was used to increase the quality of the image at night. Also the good part of this camera is their relative low price point and that they can be powered with Power over Ethernet (PoE), that is, they can be power and connected to the networking with just one cable.

TODO: add infrared image and description

12.1.2 Computer

For the computer, the main requirements where that it would need to have the necessary computing power to run the detection algorithm to detect the different vehicles and licence plates for each camera and the price point is low enough as there will be one in every community. This restiction to be able to run the detection algorithm limits to have a onboard GPU to run the models. For this, the Raspberry Pi family [15] was discarded due to the limitation of not having an onboard GPU. This leads to only be able to use a product of the NVIDIA Jetson family [16], as this are on board computers designed to be used with Machine Learning (ML) models.

For this project, the NVIDIA Jetson Nano of 4 GBs was used as it offered the best price performance ratio and was sufficient enough to perform the computing requirements.

12.1.3 Networking

In order to provide conectivity from the on-board computer to the server, a networking connection is needed to be established. As this deployment must be self sufficient and not depend on the infrastructure of each community, a router with 4G connection is deployed. For this case, a conventional router is used, the **TODO:** add router. Moreover, in order to provide the connectivity for the different cameras and the computer, a switch was used. The switch was the **TODO:** add switch.

TODO: add photo of switch and router

12.1.4 Rele Extension board

In order to trigger the mechanism to activate the opening of the garage door of each community, this is done by means of a rele connected in paralel to the door system. The opening doors function with a line where a pull-down signal is sent whenever the mechanism wants to be activated. In order to perform this operation, a Rele Extension board was used for the NVIDIA Jetson. The reason is that after trials, the on-board GPIO were not strong enought to send a constant signal for some specific communities and the mechanism malfunctioned.

The Rele Extension board used was the **TODO:** add extension due to the low price-point and the easy access to buy.

12.2 Server

The server is the point of contact to connect the users and the different communities. It allows the users to authenticate and connect to the sepcific community. It is based on a scalable archtutecture deployed in AWS due to the fast time-to-market capabilities and the scalability of it.

The server also hosts the website that the users use to connect to **TODO:** write

12.3 Monitoring

TODO: write

12.4 Deployment

TODO: write

13. IMPLEMENTATION

TODO: Write this chapter

14. TESTING

TODO: write this chapter

Part V

Results

Part VI

Conclusions

15. CONCLUSIONS

TODO: write this chapter

16. FUTURE WORK

TODO: write this chapter

17. SOCIO-ECONOMIC ENVIRONMENT

TODO: write this chapter

18. REGULATORY FRAMEWORK

The regulatory framework governing smart cities in Europe is primarily based on the data protection provisions outlined in the General Data Protection Regulation (GDPR). This is largely because these systems employ surveillance technologies such as cameras that monitor public spaces, and they handle sensitive personal data, including individuals' identification numbers and contact information. Additionally, the integration of technologies such as the IoT, Artificial Intelligence (AI), and data analytics has necessitated robust legal mechanisms to address privacy, security, and ethical considerations.

18.1 Applicable Legislation

The implementation and operation of smart city technologies in the European Union must adhere to a range of legislative measures that regulate data privacy, electronic communications, and emerging technologies. Chief among these is the GDPR, but other complementary regulations and directives are also critical in shaping the regulatory landscape.

18.1.1 General Data Protection Regulation (GDPR)

The GDPR [17], also known as the Regulation (EU) 2016/679 and enacted in 2016, provides a comprehensive framework for the protection of personal data within the European Union. Its principles of transparency, accountability, and data minimization serve as the cornerstone of privacy and data protection.

Under GDPR, smart city systems are required to obtain explicit consent from individuals before collecting or processing their personal data. Furthermore, data controllers must implement privacy-by-design and privacy-by-default principles, ensuring that data protection is an integral aspect of technological development. Smart cities must also conduct Data Protection Impact Assessments for projects involving high-risk data processing, such as large-scale surveillance or facial recognition.

18.1.2 Data Retention Policies

Smart city technologies often rely on continuous data collection through sensors, cameras, and connected devices. The GDPR mandates that data should only be retained for as long as necessary to fulfill its intended purpose. To comply, smart city operators must establish robust data retention and deletion policies. Automated systems are frequently employed to enforce these policies, ensuring that unnecessary data is securely deleted to minimize privacy risks and reduce storage costs.

18.1.3 ePrivacy Directive

The ePrivacy Directive (Directive 2002/58/EC) [18], commonly known as the "Cookie Law," governs the confidentiality of communications and the processing of electronic communications data. While initially designed for traditional telecommunications, its principles are increasingly relevant to smart city applications, such as real-time traffic monitoring and public Wi-Fi networks.

Under this directive, consent is required for the use of tracking technologies and location-based services, ensuring that users are aware of how their data is being utilized. The ongoing transition to the ePrivacy Regulation aims to enhance these protections and provide greater harmonization across member states.

18.1.4 Directive on Security of Network and Information Systems (NIS 2 Directive)

The Network and Information Systems (NIS) 2 Directive [19], adopted in 2016, focuses on the security of critical infrastructure and essential services, including those used in smart cities. It requires operators of essential services to implement risk management measures and report significant cybersecurity incidents to relevant authorities. For smart cities, this includes safeguarding critical systems such as transportation networks, energy grids, and communication systems against cyber threats.

18.1.5 Artificial Intelligence Act

The Artificial Intelligence Act [20], which came into force on 1 August 2024, establishes a common regulatory and legal framework for AI across the European Union. Its provisions will be implemented gradually over the following 6 to 36 months. The Act classifies AI systems into categories based on risk—ranging from minimal to unacceptable—and imposes specific requirements on high-risk AI applications.

For smart cities, the Act directly impacts technologies such as AI-driven surveillance systems, autonomous transportation, and automated decision-making tools. These applications must adhere to stringent requirements, including transparency, accountability, and mandatory human oversight. The Act ensures that AI technologies in smart cities are deployed ethically and align with EU values, particularly with respect to fundamental rights and safety.

18.2 Regulatory Challenges and Implications

The evolving nature of smart city technologies presents significant challenges for regulatory compliance. These challenges include the integration of multiple technologies such as IoT, AI, and data analytics, which often involve cross-border data flows and raise questions about jurisdiction and enforcement. Furthermore, the pace of technological innovation frequently outstrips the development of regulatory frameworks, creating gaps that may expose users to privacy and security risks.

Ensuring compliance with a wide array of legislative measures requires substantial investment in training and capacity-building for smart city administrators. Policymakers and industry stakeholders must collaborate to establish clear guidelines that address emerging issues while balancing innovation and the protection of fundamental rights.

18.3 Conclusion

The regulatory landscape for smart cities in Europe, underpinned by the GDPR, the ePrivacy Directive, the NIS 2 Directive, the Artificial Intelligence Act, and other complementary legislation, provides a robust foundation for ethical and secure technological deployment. As the European Union continues to refine its regulatory framework, smart cities are poised to achieve a harmonious balance between technological advancement and the protection of individual rights. By prioritizing compliance, transparency, and accountability, European smart cities can serve as global exemplars of ethical innovation.

BIBLIOGRAPHY

[1] Irene Lebrusán and Jamal Toutouh. "Assessing the Environmental Impact of Car Restrictions Policies: Madrid Central Case". In: *Smart Cities*. Ed. by Sergio Nesmachnow and Luis Hernández Callejo. Cham: Springer International Publishing, 2020, pp. 9–24. ISBN: 978-3-030-38889-8.

[2] Raquel Fernández-González et al. "Urban mobility trends and climate change: sustainability policies in the parking industry". In: *Environmental Science and Pollution Research* 30 (May 2023), pp. 1–14. DOI: `10.1007/s11356-023-26925-2`.

[3] A. Ibeas et al. "Modelling parking choices considering user heterogeneity". In: *Transportation Research Part A: Policy and Practice* 70 (2014), pp. 41–49. ISSN: 0965-8564. DOI: `https://doi.org/10.1016/j.tra.2014.10.001`. URL: `https://www.sciencedirect.com/science/article/pii/S0965856414002341`.

[4] Fredrik Dahlqvist et al. "Growing opportunities in the Internet of Things". In: *McKinsey & Company* 22 (2019).

[5] Symeon Papavassiliou Eirini Eleni Tsiropoulou John S. Baras and Surbhit Sinha. "RFID-based smart parking management system". In: *Cyber-Physical Systems* 3.1-4 (2017), pp. 22–41. DOI: `10.1080/23335777.2017.1358765`. eprint: `https://doi.org/10.1080/23335777.2017.1358765`. URL: `https://doi.org/10.1080/23335777.2017.1358765`.

[6] M. Venkata Sudhakar et al. "Development of smart parking management system". In: *Materials Today: Proceedings* 80 (2023). SI:5 NANO 2021, pp. 2794–2798. ISSN: 2214-7853. DOI: `https://doi.org/10.1016/j.matpr.2021.07.040`. URL: `https://www.sciencedirect.com/science/article/pii/S2214785321048926`.

[7] H. Al-Kharusi and I. Al-Bahadly. "Intelligent Parking Management System Based on Image Processing". In: *World Journal of Engineering and Technology* 2 (2014), pp. 55–67. DOI: `10.4236/wjet.2014.22006`.

[8] Amira A. Elsonbaty and Mahmoud Shams. "The Smart Parking Management System". In: *International Journal of Computer Science and Information Technology* 12.4 (Aug. 2020), pp. 55–66. ISSN: 0975-4660. DOI: `10.5121/ijcsit.2020.12405`. URL: `http://dx.doi.org/10.5121/ijcsit.2020.12405`.

[9] Adel Mounir Said, Ahmed E. Kamal, and Hossam Afifi. "An intelligent parking sharing system for green and smart cities based IoT". In: *Computer Communications* 172 (2021), pp. 10–18. ISSN: 0140-3664. DOI: `https://doi.org/10.1016/j.comcom.2021.02.017`. URL: `https://www.sciencedirect.com/science/article/pii/S0140366421000864`.

[10] Paul Melnyk, Soufiene Djahel, and Farid Nait-Abdesselam. "Towards a Smart Parking Management System for Smart Cities". In: *2019 IEEE International Smart Cities Conference (ISC2)*. 2019, pp. 542–546. DOI: `10.1109/ISC246665.2019.9071740`.

[11] International Council on Systems Engineering. *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. 4th. Hoboken, NJ: Wiley, 2015. URL: `https://www.incose.org/products-and-publications/se-handbook`.

[12] Alastair Ruddle. *Simplified System Engineering ‘V’ Model*. https://www.researchgate.net/figure/Simplified-system-engineering-V-model_fig1_338672391. Accessed: 2024-10-15. 2020.

[13] Rui José and Helena Rodrigues. “A Review on Key Innovation Challenges for Smart City Initiatives”. In: *Smart Cities* 7.1 (2024), pp. 141–162. ISSN: 2624-6511. DOI: 10.3390/smartcities7010006. URL: https://www.mdpi.com/2624-6511/7/1/6.

[14] General Data Protection Regulation GDPR. “General data protection regulation”. In: *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC* (2016).

[15] *Raspberry Pi*. https://www.raspberrypi.com/. Accessed: 2024-11-20.

[16] *Jetson Modules, Support, Ecosystem*. https://developer.nvidia.com/embedded/jetson-modules. Accessed: 2024-11-20.

[17] Council of European Union. *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement*. http://data.europa.eu/eli/reg/2016/679/oj. 2016.

[18] Council of European Union. *Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector*. http://data.europa.eu/eli/reg/2002/58/oj. 2002.

[19] Council of European Union. *Directive (EU) 2016/1148 of the European Parliament and of the Council of 6 July 2016 concerning measures for a high common level of security of network and information systems across the Union*. http://data.europa.eu/eli/reg/2016/1148/oj. 2016.

[20] Council of European Union. *Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence*. http://data.europa.eu/eli/reg/2024/11689/oj. 2024.