Universitá degli Studi di Messina

Dottorato di Ricerca in Ingegneria Civile Ambientale e della Sicurezza

Curriculum Scienze e tecnologie, materiali, energia e sistemi complessi
per il calcolo distribuito e le reti SSD INF01
Ciclo XXXVI

Innovative Technologies for Smart Urban Mobility Management

Author:
Francesco Martella

Head of the Doctoral School:
Prof. Dr. Gaetano Bosurgi

Advisor:
Prof. Dr. Massimo Villari

Co-advisor:
Prof. Dr. Massimo Di Gangi

Accademic Year 2022-2023

Abstract

This PhD thesis investigates the application of ICT (Information Computer Technologies) in the field of urban mobility. The objective is not to study and solve urban planning problems from a transport management or transport infrastructure point of view. The content of this PhD thesis focuses on the study of innovative digital technologies, algorithms, and systems that can be useful in the context of modern urban mobility systems both in the design and management phases. In particular, this PhD thesis reports on studies that concern the management of data from their acquisition to their processing and visualization. Digital technologies for decision support through the processing of big data were investigated, as were case studies regarding applications related to the safety of users moving in a modern urban environment. The use of technologies such as Cloud Computing, Edge Computing, and the Internet of Things (IoT) formed the basis of the studies presented. Computing resource optimization techniques have been investigated both at the Edge and at the Cloud. The decline of sensors measurement capacity was studied and a technique based on field tests was proposed to extend the devices' life. These results respond to economic and environmental sustainability problems for cities that evolve into smart cities. All the discussed aspects were investigated also taking into account cyber security issues. The techniques explained have the dual purpose of ensuring the security of the systems and the reliability of the data that provide input to algorithms and decision support systems. In general, this PhD thesis collects various research works to respond to issues relating to the use of digital technologies to support mobility in modern urban systems. The different solutions reported are based on a careful analysis of the state of the art and the goodness of the designed solutions is proven through experimental tests.

Keywords: Edge Computing, Cyber Security, IoT Rejuvenation, Smart City, Urban Mobility, Cloud Computing.

Contents

List of Figures

List of Tables

Earlier Publications

This thesis is the outcome of the doctoral degree course started three years ago. Achieved results have been published in several international conference proceedings and journals, which are listed in the following:

[1] Francesco Martella, Giovanni Parrino, Giuseppe Ciulla, Roberto Di Bernardo, Antonio Celesti, Maria Fazio, Massimo Villari. Virtual Device Model extending NGSI-LD for FaaS at the Edge. 2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid);

[2] Valeria Lukaj, Francesco Martella, Maria Fazio, Antonio Celesti, Massimo Villari. Trusted Ecosystem for IoT Service Provisioning Based on Brokering. 2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid);

[3] Alessio Catalfamo, Maria Fazio, Francesco Martella, Antonio Celesti, Massimo Villari. MuoviMe: Secure Access to Sustainable Mobility Services in Smart City. 2021 IEEE Symposium on Computers and Communications (ISCC);

[4] Agata Romano, Rosaria Lanza, Fabrizio Celesti, Antonio Celesti, Maria Fazio, Francesco Martella, Antonino Galletta, Massimo Villari. Towards Smart Tele-Biomedical Laboratory: Where We Are, Issues, and Future Challenges. 2021 IEEE Symposium on Computers and Communications (ISCC);

[5] Lorenzo Carnevale, Armando Ruggeri, Francesco Martella, Antonio Celesti, Maria Fazio, Massimo Villari. Multi Hop Reconfiguration of End-Devices in Heterogeneous

Edge-IoT Mesh Networks. 2021 IEEE Symposium on Computers and Communications (ISCC);

[6] Antonino Quattrocchi, Damiano Alizzio, Francesco Martella, Valeria Lukaj, Massimo Villari, Roberto Montanini. Effects of Accelerated Aging on the Performance of Low-Cost Ultrasonic Sensors Used for Public Lighting and Mobility Management in Smart Cities. Sensors 2022, MDPI;

[7] Francesco Martella, Giovanni Parrino, Mario Colosi, Giuseppe Ciulla, Roberto Di Bernardo, Marco Martorana, Roberto Callari, Maria Fazio, Antonio Celesti, Massimo Villari. URBANITE: Messina Use Case in Smart Mobility Scenario. 24th International Multiconference Information Society;

[8] Giuseppe Ciulla, Roberto Di Bernardo, Isabel Matranga, Francesco Martella, Giovanni Parrino, Shabnam Farahmand. How Disruptive Technologies can Strengthen Urban Mobility Transformation.The Experience of URBANITE H2020 Project. 24th International Multiconference Information Society;

[9] Mario Colosi, Francesco Martella, Giovanni Parrino, Antonio Celesti, Maria Fazio, Massimo Villari. Time Series Data Management Optimized for Smart City Policy Decision. 2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid);

[10] Valeria Lukaj, Francesco Martella, Antonio Celesti, Maria Fazio, Massimo Villari. An Enriched Visualization Tool based on Google Maps for Water Distribution Networks in Smart Cities. 2022 IEEE Symposium on Computers and Communications (ISCC);

[11] Francesco Martella, Maria Fazio, Antonio Celesti, Valeria Lukaj, Antonino Quattrocchi, Massimo Di Gangi, Massimo Villari. Federated Edge for Tracking Mobile Targets on Video Surveillance Streams in Smart Cities. 2022 IEEE Symposium on Computers and Communications (ISCC);

[12] Valeria Lukaj, Francesco Martella, Antonino Quattrocchi, Maria Fazio, Roberto Montanini, Massimo Villari, Antonio Celesti. Towards IoT Rejuvenation: a Study on HY-SRF05 Ultrasonic Sensor Ageing for Intelligent Street Pole Lamp Control in a Smart City. 2022 IEEE Symposium on Computers and Communications (ISCC);

[13] Valeria Lukaj, Francesco Martella, Maria Fazio, Armando Ruggeri, Antonio Celesti, Massimo Villari. A Blockchain Based Federated Ecosystem for Tracking and Validat-

ing the Authenticity of Goods. 2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech);

[14] Francesco Martella, Maria Fazio, Giuseppe Ciulla, Roberto Di Bernardo, Antonio Celesti, Valeria Lukaj, Mario Colosi, Massimo Di Gangi, Massimo Villari. An Edge System for the Safety of Cyclists in the Urban Area. 2022 IEEE International Smart Cities Conference (ISC2);

[15] Valeria Lukaj, Christian Sicari, Francesco Martella, Antonio Celesti, Maria Fazio, Massimo Villari. An Innovative Method for 3D Virtual Indoor Navigation based on Geotags. 2030 d.c. Proiezioni future per una progettazione sostenibile;

[16] Alessio Catalfamo, Lorenzo Carnevale, Antonino Galletta, Francesco Martella, Antonio Celesti, Maria Fazio, Massimo Villari. Scaling Data Analysis Services in an Edge-based Federated Learning Environment. 2022 IEEE/ACM 15th International Conference on Utility and Cloud Computing (UCC);

[17] Valeria Lukaj, Francesco Martella, Maria Fazio, Antonio Celesti, Massimo Villari. Establishment of a trusted environment for IoT service provisioning based on X3DH-Based brokering and Federated Blockchain. Internet of Things, Elsevier.

[18] Francesco Martella, Valeria Lukaj, Maria Fazio, Antonio Celesti, Massimo Villari. On-Demand and Automatic Deployment of Microservice at the Edge Based on NGSI-LD. 2023 31st Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP).

[19] Valeria Lukaj, Francesco Martella, Maria Fazio, Antonino Galletta, Antonio Celesti, Massimo Villari. Gateway-Based Certification Approach to Include IoT Nodes in a Trusted Edge/Cloud Environment. 2023 23nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid).

[20] Valeria Lukaj, Alessio Catalfamo, Francesco Martella, Maria Fazio, Massimo Villari, Antonio Celesti. A NoSQL DBMS Transparent Data Encryption Approach for Cloud/Edge Continuum. 2023 IEEE Symposium on Computers and Communications (ISCC).

[21] Cihan Sakman, Panagiotis Gkikopoulos, Francesco Martella, Massimo Villari, Josef Spillner. Indoor Navigation for Personalised Shopping: A Real-Tech Feasibility Study.

20th International Conference on Smart Business Technologies - ICSBT;

[22] Antonino Quattrocchi, Damiano Alizzio, Francesco Martella, Valeria Lukaj, Massimo Villari, Roberto Montanini. Valutazione degli Effetti dell'Invecchiamento accelerato sulle Performance di Sensori Low-Cost di Posizione Impiegati in Ambito Smart Cities - V Forum Nazionale delle Misure 2021 Taormina 09/2021;

[23] Antonino Quattrocchi, Francesco Martella, Valeria Lukaj, Rocco De Leo. Massimo Villari, Roberto Montanini. Designing a Low-Cost System to Monitor the Structural Behavior of Street Lighting Poles in Smart Cities. Sensors, MDPI;

[24] Armando Ruggeri, Antonio Celesti, Valeria Lukaj, Francesco Martella, Ilenia Celesti, Maria Fazio, Massimo Villari. Prospettive sull' Utilizzo dello "Smart Contract" a Supporto della Continuità Territoriale per la Prenotazione Dinamica di Viaggi. Editoriale Scientifica;

CHAPTER 1

Introduction

Modern urban systems are evolving towards new models of sustainable mobility based on the use of Intelligent Transport Systems (ITS) and infomobility systems on the one hand, and intelligent solutions for sharing resources (e.g., bike sharing, scooter sharing, etc) on the other. This evolution certainly has technological aspects, but also sociological and cultural ones. Public Administrations, Local Public Transport (LPT) companies but also private companies offer new services to citizens in the field of urban mobility. These services concern info mobility, transport booking, sharing, goods mobility, tracking of objects, facility for purchases, etc. The services, through the applications that supply them, generate large quantities of data, which are widely used for the development of decision support systems for public administrations and public or private transport companies.

As already mentioned, this evolution consists of several technological components and therefore of a series of problems related to their use. In fact, innovative mobility systems integrate heterogeneous technological solutions that operate at different levels:

- Environment, vehicles and people monitoring: IoT (Internet of Things), Edge devices;
- Storage, management and processing of Big Data: Cloud computing, relational and NoSQL storage systems, Big Data Analysis, etc;
- Algorithms for decision support and service optimization: Machine Learning, Business Intelligence;

- Communication and decision support systems: web-oriented info mobility platforms, decision dashboards;

The technologies mentioned must ensure an efficient, scalable, and sustainable mobility system, including identity management mechanisms not only for users, but also for the devices used, scalable and secure tracking of activities, data protection, and reliability.

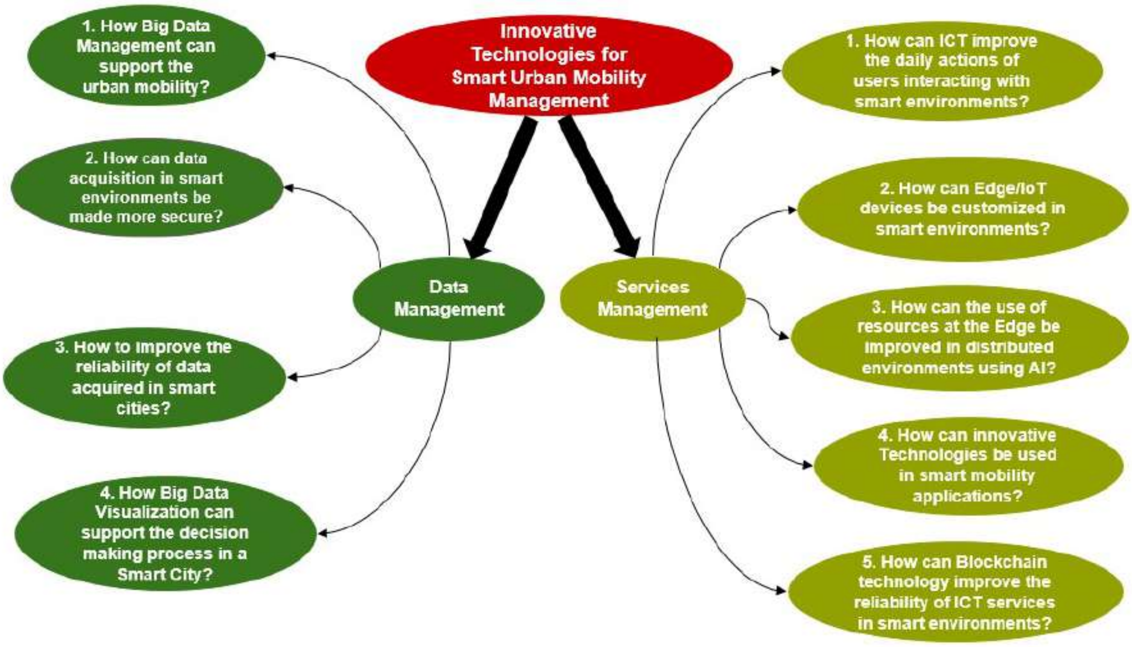

Innovative Technologies for Smart Urban Mobility Management
Data Management
Services Management
1. How Big Data Management can support the urban mobility?
2. How can data acquisition in smart environments be made more secure?
3. How to improve the reliability of data acquired in smart cities?
4. How Big Data Visualization can support the decision making process in a Smart City?
1. How can ICT improve the daily actions of users interacting with smart environments?
2. How can Edge/IoT devices be customized in smart environments?
3. How can the use of resources at the Edge be improved in distributed environments using AI?
4. How can innovative Technologies be used in smart mobility applications?
5. How can Blockchain technology improve the reliability of ICT services in smart environments?

Figure 1.1: Research Questions (RQ) Overview.

This chapter describes the main concepts and characteristics useful for satisfying the described requirements. The research questions investigated were grouped into specific topics (Figure 1.1) as follows:

Data management:

- **Research Question 1 (RQ1):** How Big Data Management can support the urban mobility?
- **Research Question 2 (RQ2):** How can data acquisition in smart environments be made more secure?
- **Research Question 3 (RQ3):** How to improve the reliability of data acquired in smart cities?
- **Research Question 4 (RQ4):** How Big Data Visualization can support the decision-making process in a Smart City?

Service Management:

- **Research Question 1 (RQ1):** How can ICT improve the daily actions of users interacting with the smart environments?
- **Research Question 2 (RQ2):** How can Edge/IoT devices be customized in smart environments?
- **Research Question 3 (RQ3):** How can the use of resources at the Edge be improved in distributed environments using AI?
- **Research Question 4 (RQ4):** How can innovative Technologies be used in smart mobility applications?
- **Research Question 5 (RQ5):** How can Blockchain technology improve the reliability of ICT services in smart environments?

1.1 Scientific contributions

In this Section, the scientific contributions produced to answer to the research questions are presented.

Data Management:

Figure 1.2 reports the association of each research question with the scientific contribution listed in Section Earlier Publications.

Research Question (RQ1): How Big Data Management can support the urban mobility?

Contribution 1: In Smart Cities, it is essential to pay attention to issues relating to urban mobility. Today Smart Mobility allows people to optimize their journeys by reducing the stress associated with them, while Sustainable Mobility helps protect the environment by improving the quality of life in Smart Cities. The contexts of urban mobility, but not only, require today more than ever the use of large amounts of historical data to carry out the necessary analyses for the various use cases. Therefore, good management of time series data becomes indispensable, able to consider in an optimized way the concepts of layout and to provide the user with specification functions. This need has been incorporated as a native implementation in some Database Management Systems (DBMS). However, considering the state of digital transformation in public administrations (PA), it was necessary to ask how to

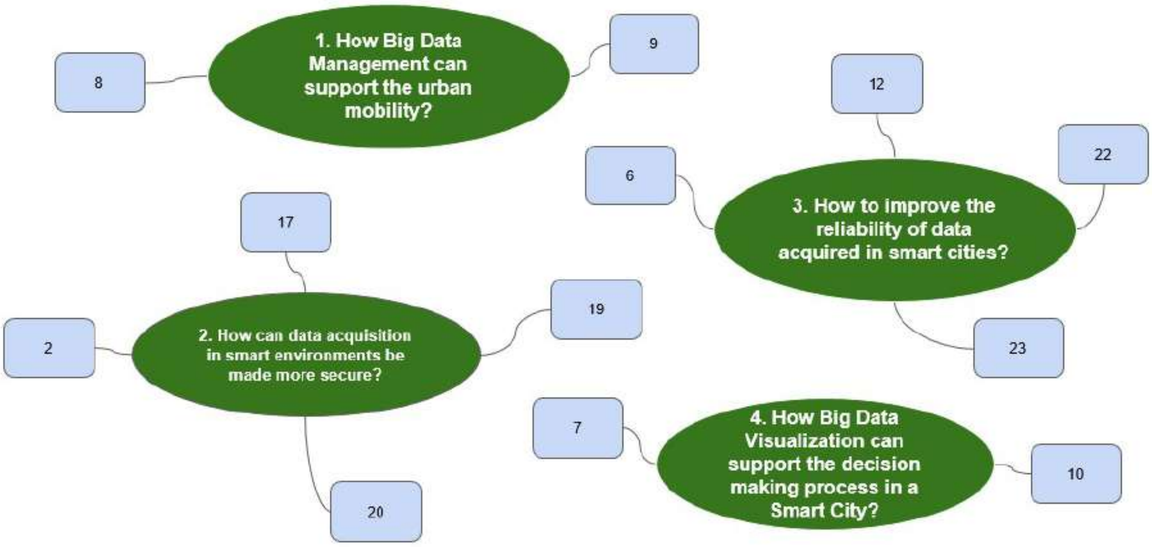

1. How Big Data Management can support the urban mobility?
8
9
12
22
6
3. How to improve the reliability of data acquired in smart cities?
17
2
2. How can data acquisition in smart environments be made more secure?
19
23
7
4. How Big Data Visualization can support the decision making process in a Smart City?
10
20

Figure 1.2: Data Management area scientific contribution.

find a solution for previous versions of DBMSs. A general solution makes it possible to use the Big Data available to the PAs in an optimized way, arriving at more efficient solutions in a short time. Details for this contribution with references to real use cases have been addressed in [8, 9].

Research Question 2 (RQ2): How can data acquisition in smart environments be made more secure?

Contribution 2: The adoption of uncertified Internet of Things (IoT) devices can expose the system to cyber attacks that can disrupt IoT-based applications or generate fake data. At the same time, the massive use of Edge Computing and Internet of Things (IoT) devices can cause the growth of a large number of services affected by errors and exposed to various cyber-attacks. The data exchanged with the cloud infrastructure to offer the service to the end user, in fact, exposes the systems to these risk scenarios. To address these challenges, a new broker-based certification process was created that decouples the communication between IoT devices and the Certification Authority (CA) at the edge. Acting as an "Intermediary", the Mobile Edge Computing (MEC) node secures the communication between untrusted IoT devices and the CA, taking responsibility for node certification. Creating a trusted ecosystem is further strengthened to ensure data integrity and non-repudiation using a federated Blockchain. Encrypting data at the Edge and exchanging encrypted data can also be a solution to the proposed challenges or further strengthening of the proposed solutions. Details for this contribution with references to real use cases have been addressed in [2, 17, 19, 20].

Research Question 3 (RQ3): How to improve the reliability of data acquired in smart cities?

Contribution 3: In the field of Smart Cities and even Smart Mobility, the use of low-cost devices is considered an advantageous solution due to their easy availability, cost reduction, and, consequently, technological and methodological development. However, this type of transducer shows many critical issues, e.g., in metrological and reliability terms, which can significantly compromise their functionality and safety. It is important to evaluate the effects of sensor aging for a better understanding of the phenomena. It is possible to accelerate sensor aging in extreme climatic conditions and evaluate the performance of a control system, based on low-cost sensors. This is the starting point for the concept of IoT Rejuvenation, a proactive cost-effective technique that can contrast the inevitable ageing of IoT systems guaranteeing the accuracy of collected data over time. Details for this contribution with references to real use cases have been addressed in [6, 12, 22, 23].

Research Question 4 (RQ4): How Big Data Visualization can support the decision-making process in a Smart City?

Contribution 4: The urban transformations and the changes that the world is going through lead, today more than ever, to the need to make quicker and more timely choices in the field of mobility management. Technology is therefore essential to provide decision support tools that help managers and policymakers to better manage cities. In this context, new technologies and data visualization methods can be of support in the field of urban mobility. To meet these challenges, software tools are needed that, starting from a set of local data regarding traffic and public transport tracking, allow technicians to quickly visualize the state of traffic or bottlenecks for public transport on a map. Of particular importance are therefore the pre-processing of the data (data deriving from surveys, geo-referencing of the acquisition systems, etc.) and their visualization on commonly used tools. Details for this contribution with references to real use cases have been addressed in [7, 10].

Services Management:

Figure 1.3 reports the association of each research question with the scientific contribution listed in Section Earlier Publications.

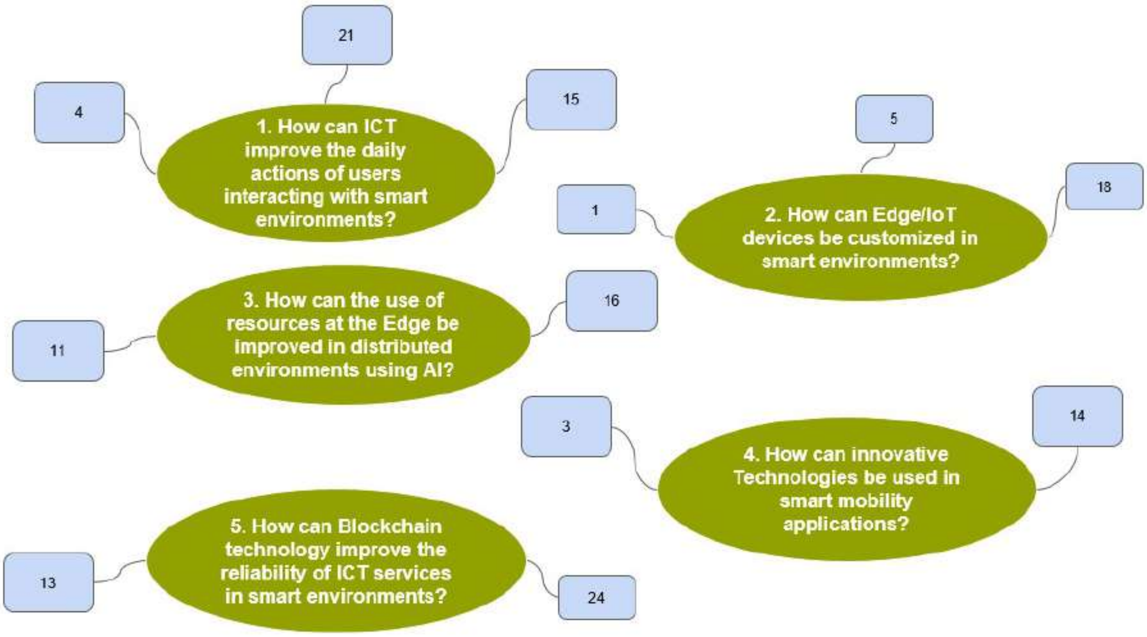

21
4
15
1. How can ICT improve the daily actions of users interacting with smart environments?
5
1
18
2. How can Edge/IoT devices be customized in smart environments?
3. How can the use of resources at the Edge be improved in distributed environments using AI?
16
11
3
14
4. How can innovative Technologies be used in smart mobility applications?
5. How can Blockchain technology improve the reliability of ICT services in smart environments?
13
24

Figure 1.3: Services Management area scientific contribution.

Research Question 1 (RQ1): How can ICT improve the daily actions of users interacting with smart environments?

Contribution 1: Moving around cities using public transport requires the need to plan the times necessary for the various daily actions. The introduction of ICT technologies in everyday life makes it possible to improve the daily actions of the user who interacts with smart environments. Thanks to specific user applications and broadband connectivity it is possible to carry out certain actions remotely or in any case to use advanced technologies to optimize movements both inside and outside buildings. These applications support users in real time both through the use of sensors and IoT technologies, either through information collected through social networks or in general on the network from other users. The efficient and effective management of data and the design of applications that use specific sensors, even at low cost, allow for the creation of applications, for example for telemedicine or indoor navigation. Details for this contribution with references to real use cases have been addressed in [4, 15, 21].

Research Question 2 (RQ2): How can Edge/IoT devices be customized in smart environments?

Contribution 2: The Internet of Things has revolutionized the way services are delivered in intelligent environments. However, IoT devices have limited resources, and reconfiguring them can be very difficult and costly. In these cases, Edge Computing represents a solution to support the IoT with flexible resource management. It is possible to think of methods to automate the firmware update but also the type of service hosted on the device. It is possible to use the same sensor to provide different services to the user also in terms of data processing and sampling. For this, however, it is necessary to standardize methods for an automatic and flexible configuration. For this purpose, it is possible to use data models with a flexible structure that can be adapted to different needs. By introducing high abstraction data of the service on a specific device, it is possible to "virtualize" the device (virtual device) and therefore virtually replicate the physical component making the most of the resources. For this purpose, the key characteristics of a smart environment have been identified and the concept of a virtual device has been defined, i.e. an abstract component characterized by specific high-level functions. Furthermore, a software architecture has been defined that allows the described concepts to be implemented. Details for this contribution with references to real use cases have been addressed in [1, 5, 18].

Research Question 3 (RQ3): How can the use of resources at the Edge be improved in distributed environments using AI?

Contribution 3: Smart environments include different types of edge devices and sensors. Technological progress brings the Public Administrations but also companies to install new devices with different technology. The consequence of these situations is to find cities full of different devices with different technologies that perform the same function. In order to optimize the use of these resources, also with a view to sustainability, the concept of Federated Edge was used. The concept behind this principle is that devices, even using artificial intelligence, can collaborate by communicating with each other. For example, through video surveillance systems it may be possible to follow a target with specific devices that can better perform this operation rather than interrogating a video stream in a specific area framed by different devices at different angles. Details for this contribution with references to real use cases have been addressed in [11, 16].

Research Question 4 (RQ4): How can innovative Technologies be used in smart mobility applications?

Contribution 4: Urban areas are evolving towards mobility management systems based on sustainable mobility. Technologies such as Big Data, Cloud/Edge Computing and IoT are fundamental in this evolution. In fact, it is necessary to guarantee secure data exchange in order to give users reliable services which guarantee their safety. Furthermore, it is essential to create systems that, in addition to providing services to users, feed the databases of the Public Administrations, or service providers so that they can better implement and use decision-support services. Indeed, the efficiency of these services depends on the quality and quantity of data acquired. In this context, it was interesting to analyze case studies related to the use of e-bikes in smart environments. On the one hand, a use case concerning security technologies in the PA was analyzed for the purpose of providing services to users, on the other hand, a case study of a device capable of collecting data for users and the PA was analyzed to provide services to users. Details for this contribution with references to real use cases have been addressed in [3, 14].

Research Question 5 (RQ5): How can Blockchain technology improve the reliability of ICT services in smart environments?

Contribution 5: Blockchain technology represents one of the most innovative trends in various fields of ICT application. In fact, it opens up new scenarios in the field of the Internet of Transactions thanks to seven main characteristics: decentralization, security, consensus, immutability, transparency, accountability and programmability. One of its application domains is represented by the smart contract, an IT protocol aimed at facilitating, verifying and digitally forcing the negotiation of an agreement between subjects without the need for third-party certification. Therefore, the Blockchain is one of the most promising technologies in the legal field. A notable application innovation has been defined as Federated Blockchain. The verification of Smart Contracts stipulated on different Blockchain systems can help to counteract phenomena such as the counterfeiting of a good or a service. This concept is based on the use of "Private" Blockchains managed by service providers for the certification of their goods, and of "public" Blockchains where agreements between users and suppliers are certified. Details for this contribution with references to real use cases have been addressed in [13, 24].

1.2 Structure of the Thesis

Figure 1.4 shows the general organization of the thesis maintaining the color scheme used in the 2 topics described previously. The purpose of Figure 1.4 is to highlight the interaction between the research questions, the scientific articles and the corresponding chapters within the thesis.

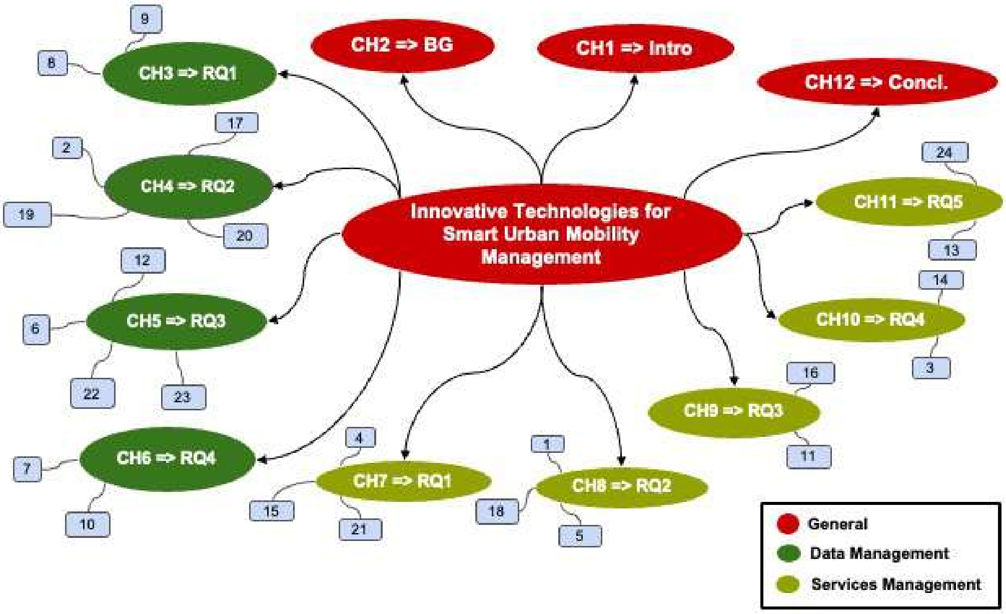


Figure 1.4: Thesis General Overview.

Detailed contributions to the research questions (RQs) are reported in subsequent chapters. Below is a brief description:

- Chapter 2 summarizes a brief description of the basic technologies used in this PhD thesis.

- Chapter 3 contains the contribution for RQ1 of the Data Management topic. In particular, this chapter focuses on functional applications of Big Data analysis in urban mobility scenarios. The application of advanced bucketing methods for the optimization of large databases to be analyzed to provide decision-makers with decision support software is reported. Furthermore, possible applications for decision-making in urban mobility are described.

- Chapter 4 reports the contribution for the RQ2 of the Data Management topic. Chapter 4 describes methods for certifying IoT devices in untrusted environments. In detail, it explains how to create a trusted ecosystem for IoT devices and how to use Edge devices for the certification of IoT nodes in the trusted environment. The described systems are useful for sensors that can be used in remote areas for example in case of mobility infrastructure monitoring.

- Chapter 5 contains the contribution for RQ3 of the Data Management topic. In Chapter 5 we focused on issues relating to data reliability. Considering the problem of economic sustainability for smart cities, it is clear, even in the specific case of mobility, that it is not sustainable to use expensive devices that must be continuously maintained. For this purpose, IoT applications in urban scenarios are described and the IoT Rejuvenation technique is defined.

- Chapter 6 reports the contribution for the RQ4 of the Data Management topic. Chapter 6 focused on Big Data visualization methods used both in the mobility field and the smart city field in general. This chapter reports real use cases in the mobility field. The described application was used by the Municipality of Messina in a European project. Furthermore, the use case incorporates the querying methods described in chapter 3. An algorithm for converting geo-referenced data from a generic X, Y system to a WGS-84 reference system is described.

- Chapter 7 contains the contribution for RQ1 of the Services Management topic. This chapter reports use cases related to the case of managing user movement within smart buildings. Use cases of applications that support users inside buildings to reduce product search times are described. Furthermore, the usefulness of telematic systems is underlined in order to avoid the movement of users or provide essential services even in the case of conditions that limit the users' mobility.

- Chapter 8 reports the contribution for the RQ2 of the Services Management topic. Chapter 8 to meet the needs of services in smart cities, including those of smart urban mobility, describes a solution for the on-demand deployment of services on Edge/IoT devices. The use of the NGSI-LD protocol is designed and tested to define a method for the automatic on-demand deployment of microservices to reduce the number of physical devices and optimize computing resources. This aspect is also linked to the environmental and economic sustainability of smart cities.

- Chapter 9 contains the contribution for RQ3 of the Services Management topic. Edge/IoT devices collect and process large amounts of data. This chapter focuses on the concept of federation to optimize the functioning of applications. The design of a Federated

Edge application for the optimization of video surveillance services is described. Furthermore, the use case of a federated learning application optimized via Edge devices is reported.

- Chapter 10 reports the contribution for the RQ4 of the Services Management topic. Chapter 10 describes mobility services in urban areas. The case studies reported refer to a platform for the long rental of e-bikes for public administration and to a device that interfaces with a public administration cloud to increase the safety of bicycles. The two case studies refer to cloud services in use in the municipality of Messina for National and European projects.
- Chapter 11 contains the contribution for the RQ5 of the Services Management topic. This chapter describes possible uses of the Blockchain, highlighting its usability in smart city scenarios. The chapter concludes with a focus on smart contracts to be used in the field of urban mobility. A use case is analyzed in which a user must buy several tickets to go from a departure to a destination.
- Chapter 12 concludes the thesis with a final remark and highlights for future research.

Each chapter is motivated by trying to propose solutions for the highlighted issues. The proposed solutions have not already been identified in the state of the art for the reported issues. Both problem analysis and the design of innovative solutions are described. Where the designed solutions have been implemented, experimental tests support the arguments of the thesis.

CHAPTER 2

Background

This chapter provides background information about technologies used as the basis of this thesis. The contents are general only for a better understanding of the presented works.

2.1 Cloud Computing

Cloud Computing is a technology that consists of the distribution of computing services through the Internet (“cloud”). In general, it can be defined as the provision of online computing services demand, starting from the infrastructure for data management up to the application, in pay-to-go. Cloud Computing guarantees the availability of the resources of a generic information system to the users or entities that request it. These infrastructures can provide systems for data storage known as cloud storage, and computing power without direct active management by the user. In general, the term cloud computing refers to data centers that provide high-performance services to numerous users via the Internet. The Large clouds perform functions distributed on different central servers, in particular, if there is a relatively close connection it can be defined as a server perimeter. Cloud systems can be configured for a single organization in the case of corporate clouds or be available to numerous organizations, or public cloud. Cloud Computing is based on sharing of resources with the aim of providing high-performance services in large economies of scale. The use of public or hybrid cloud solutions allows you to reduce or completely avoid costs for the physical creation of the entire IT infrastructure. Also, an application running on an infrastructure Cloud turns out to be faster and more performant, with better manageability and less maintenance. These features allow service providers to tailor resources based on the requests made by users, to satisfy the demand by integrating the burst computing capacity, i.e. high computing power in certain periods of time in which the demand for resources reaches a peak. Cloud service providers use a payment model based on the resource consumption of a general user. The development of this technology in recent years is linked to the availability of high-capacity networks, hardware resources, and low-cost storage devices. Cloud services are based on hardware virtualization systems and service-oriented architectures for performing autonomous computing operations. In these infrastructures, the Cloud Service Provider (CPS) proceeds with the control, maintenance, and data collection operations on the firewalls. In addition, intrusion identification services or systems that allow them are implemented to perform actions against attacks on information flows within a generic network.

2.1.1 Cloud models

There are many cloud service providers today. Any of these services can be categorized according to the features and functionality it offers. It is possible to make one first distinction based on the cloud computing model adopted :

- IaaS (Infrastructure as a Service): refers to online services that provide high-end APIs level, applicable for low-level information extraction of network infrastructure such as processing, scalability, security, and data backup resources. The hypervisor proceeds with running virtual machines as guests and can support and manage large numbers of services according to user requests;
- PaaS (Platform as a Service): these are models in which cloud service providers provide a computing platform including the operating system, databases, and web servers. Typically the toolkits and standards for the development, distribution channels of the services, and payment management. Some Paas providers organize the resources of storage and automatically resize them according to application requests, in such a way that the end user does not proceed with the manual allocation of the same;
- Saas (Software as a Service): are the models in which users have access to the software application and storage systems. Cloud service providers proceed with the management of software services based on user requests. The services are provided by adopting payment policies based on consumption and a tariff for the subscription. Generally,

these models are configured and managed by cloud application software, placing increased focus on scalability with task cloning operations on different virtual machines. For the management of a large number of requests from users, Cloud applications must be multi-tenant and serve multiple user organization systems.

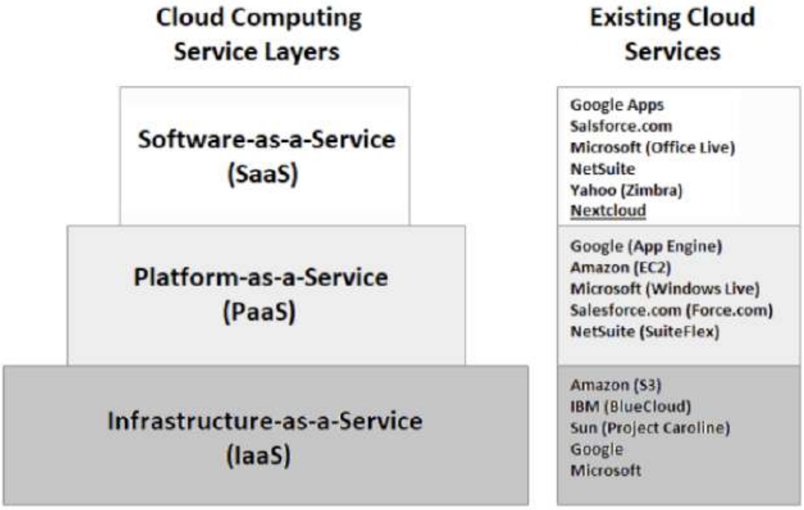


Figure 2.1: Cloud Computing Service Layers.

For each model there are different services available. An example, only for representation, is shown in Fig. 2.1,

2.1.2 Cloud Typology

In general, the choice of how to deploy cloud services depends solely on the organizations and the needs of the entities that have to use them. Exist different Cloud typologies:

- Public: This is the most common Cloud Computing solution. In Public Clouds, resources such as servers and storage, are managed by the proprietary provider and provided through an Internet connection. All hardware, software, and other infrastructure supporting the services clouds are owned by the vendor provider and are shared with cloud tenants. In general deployment of public cloud services is for the provision of e-mail, or archiving and documentation applications.

- Private: Cloud infrastructure configured for a single organization and managed by a third party. For the creation of a Private Cloud, it is necessary to define the processes of virtualization of the corporate environment that require the presence of external resources. These systems are expensive as a physical data center is required for each cloud privacy and a security system for sensitive data.

- Hybrid: It is a mix of Public and Private Cloud infrastructure. Generally, this solution has the advantage of offering multiple deployment models and can connect managed or dedicated collocation services with cloud resources. These systems also allow you to exceed the limits of a provider so that it cannot be placed in a single category of Cloud as Public or Private. Integrating the concepts of aggregation, integration, and personalization of the services it is possible to extend the management capabilities of the services cloud. Through this infrastructure, an organization can store data directly from sensitive users of a private cloud application and interconnect them with applications of business intelligence provided by public clouds.

2.2 Internet of Things

The Internet of Things (IoT) refers to a new reality in which intelligent objects connected to the Internet interact with human beings, give information and receive information, and commands to perform specific tasks automatically. This term indicates a family of technologies that have the purpose of making any object, even if conceived, designed, and created without predisposition for digital use, a device connected to the Internet capable of enjoying all the characteristics that objects conceived, designed, and made to use the network. An "object" belongs to the IoT world if it mainly can be used for:

- Monitoring: the object must behave as a sensor, it must be capable of producing information about itself and/or the surrounding environment. An example can be a smart lamp designed in the IoT field. It can signal its operating status, its position, and the energy it is consuming, but it can also signal the operating status of the electricity grid that feeds it or the meteorological situation in its surroundings;
- Control: the object can be controlled remotely in a simple way through the use of a special interface that allows communication with the object using the Internet. For example, if the IoT entity is a solenoid valve that controls a water network junction or a smart water meter, it must be possible to open and close it remotely in real-time;

The main characteristics of the IoT are as follows:

- Inter connectivity: for the IoT, everything can be interconnected, with information and communication as fundamental elements;

- Services related to things: the IoT is able to provide services to things, within limits, respecting the protection of privacy and distinguishing physical objects from virtual ones;
- Heterogeneity: IoT devices are heterogeneous, they can be built on different hardware and network platforms;
- Dynamic change: the state of the devices can change rapidly, go from an active to a waiting state, from connected to disconnected, and change their position or their processing speed. Also, the number of devices can change dynamically;
- Security: to make the most of the benefits of the IoT, security is a key element. Whoever uses this technology must take into account the security of the data that travels on the network and is collected by these devices.
- Connectivity: Connectivity allows accessibility to the network.

2.2.1 IoT Architecture

IoT architecture is made up of different layers (Fig. 2.2). This conformation allows, as you go up to decrease the complexity of the technology.

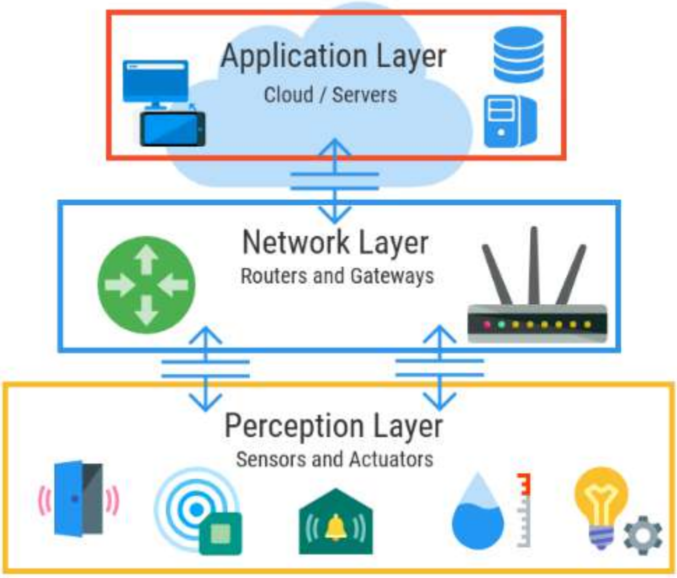


Figure 2.2: IoT Layers

In addition, it allows you to understand the relationship between the fundamental tools that make up the IoT:

- Perception layer: the lowest level consists of all those devices capable of detecting data from the physical world and converting it into digital. Sensors are a tool capable of doing this and can be grouped according to their purpose: environmental, for the home, for the body, for household appliances, etc...;
- Network layer: the intermediate layer is the one that concentrates and manages the data retrieved from the perception layer. Data analysis, management, security control, and data routing operations are performed in this layer;
- Application layer: The top layer is the application layer. It is the layer with which the real contact with the user takes place. It allows to hide the complexity of the lower layers and to make environments in the field of construction, transport, city, personal life, and many other "smart" areas.

2.2.2 IoT Security

The security of IoT devices is one of the most important aspects to consider when designing ICT systems for Smart Cities. Also for IoT devices, it is necessary to preserve *Confidentiality*,*Integrity*, and *Availability* (CIA), but there are problems in applying the classic techniques on IoT devices.

IoT devices are limited from the hardware point of view and for this reason they cannot take advantage of all the protection and security mechanisms that can be implemented on systems equipped with more performing hardware. Furthermore, devices have power, connectivity, and scalability needs that imply communication authentication mechanisms, integrity, and end-to-end security.

When an IoT device is activated, it must *authenticate* itself in the network before exchanging data. Once connected, given the reduced computational capabilities, it is necessary to filter packets addressed to the device and manage software updates.

As far as *Confidentiality* is concerned, it is essential to guarantee the availability and accessibility of data only to authorized users. To guarantee the *Integrity* of the acquired and transmitted data it is necessary to use mechanisms that validate and certify the data source, i.e. the IoT, and adopt secure transmission protocols that guarantee end-to-end communication. The *availability* of IoT systems is guaranteed when all services and applications are reachable when required by the user.

A generic IoT device connected to the ICT system of a given organization can be subject to different cyber attacks that can affect its operation, such as DDoS (Distributed Denial of

Service), or use the device as an access point to the network to steal sensitive information and run a phishing attack inside. As previously reported in Figure **??** the architecture of IoT systems is characterized by three levels, each with its security requirements: Perception Layer, Network Layer, and Application Layer.

- **Security in Perception Layer**: In this layer, the devices are connected wirelessly. A possible vulnerability is the presence of other signals that can intercept the wireless signal of the IoT device. Many IoT devices use RFID (Radio Frequency Identification) technology and are equipped with a very small storage capacity and minimal processing capacity, thus revealing themselves to be very vulnerable to different attacks. For example, confidentiality at this level can be compromised by Replay Attacks by spoofing a device's identity or by breaching encryption keys via timing attacks. At this level, robust encryption, authentication, and access control mechanisms must be applied to counter attacks.
- **Security in Network Layer**: At this level, there are different attacks present on the Internet. In addition to DDOS attacks, communications between IoT nodes can be intercepted and sniffed or be subject to Man-in-the-Middle attacks that can intercept communications by violating encryption keys and compromising the security of the communication channel.
- **Security in Application Layer**: At this level, IoT devices are vulnerable to a possible system overload that can compromise the availability of services.

2.3 Edge Computing

With the diffusion of distributed IoT systems and Cloud-based services optimized for Smart Cities ,the need for new methods of processing data acquired from a large number of sensors has arisen.

Currently, ICT systems are evolving towards based models on the Edge Computing paradigm, the adoption of which makes it possible to reduce the costs and resources necessary for data processing on classic Cloud systems. The Edge Computing paradigm has been used to define a new form of processing that is performed on-site or near a particular data source, reducing the need to process data in a remote data center in the Cloud.

Compared to traditional forms of computing, Edge computing offers organizations a faster and more efficient way to process data that is produced at the edge of the network. In the past, Edge Nodes generated large amounts of data that was often not used. With the spread of distributed and decentralized IT architectures with Mobile Computing and IoT, it is possible to obtain real-time information with less latency and lower bandwidth demands of the cloud server.

Edge Computing is a more efficient solution than cloud computing for different services. Going to allocate all data processing services to the Cloud is an efficient way of processing data. But with the growing amount of data being generated by Edge Devices, data transport speed is becoming the bottleneck for the Cloud. If all captured data were sent to the Cloud for processing, the system's response time would be too long. Also, current network bandwidth and reliability would not be able to support the large amount of data in transit. A possible solution to these problems is the adoption of Edge solutions where response times are shorter, processing is more efficient and there is less pressure on the network.

In Edge solutions, most of the data produced by the IoT is not transmitted to the Cloud but is processed at the edge of the network. Furthermore, IoT nodes with wireless connectivity require a lot of energy and therefore processing data at the edge is an optimal solution from an energy point of view. In general Edge computing refers to the enabling technologies that allow calculations to be performed at the edge of the network, on downstream data on behalf of cloud services, and upstream data on behalf of IoT services. "edge" is any computing and network resources along the path between data sources and cloud data centers.

2.3.1 Edge System General Architecture

As previously described Edge Computing takes advantage of computing resources outside the data center. This process is necessary to move the workload closer to where the data is created in order to respond faster to analyzes. A generic Edge architecture is schematized in Fig. 2.3 (https://www.lfedge.org/2020/03/05/edge-computing-architecture-and-use-cases/).

Here are some key elements of the architecture:

- Cloud: It can be public or private. The cloud also hosts and runs the applications used to orchestrate and manage the different edge nodes. Additionally, cloud and edge workloads can balance each other. In case of specific applications the cloud, without losing its nature, can be a source and a destination for all the data requested by the other nodes;

- Edge device: it is a hardware component equipped not only with sensing or actuation

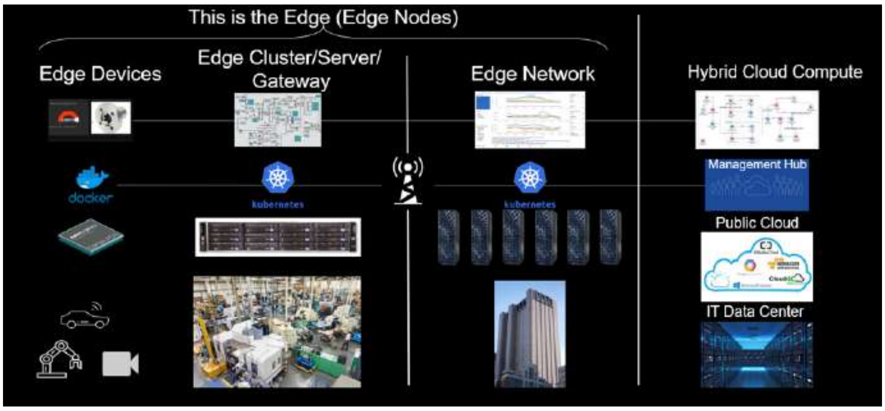


Figure 2.3: Edge Architecture

capabilities but also with processing capabilities integrated into the device. However, an edge device has limited computing resources. These are devices equipped with ARM or x86 class CPUs with 1 or 2 cores, 128 MB of memory, and limited local persistent memory (4 GB);

- Edge node: is a generic edge device, edge server, or edge gateway. This is the name of a component of the architecture that can have multiple functions;
- Edge server/cluster: is a general IT computer located at the location of data collection or at the location where command implementation takes place. This is usually an industrial PC or rack computer. An edge server/cluster can be used in complex architectures where shared resources are required in geographically close environments;
- Edge gateway is typically an edge server/cluster that, in addition to being able to host specific application workloads, can also function as a network device;

In this architecture, sensors are equipment that collects and transmits data to an edge/cloud but lack advanced computing resources. IoT devices are not considered in Edge architectures because they have pre-established functions that do not influence the architecture.

2.4 Data Model

Data modeling can be defined as a form of data governance. This definition refers to the definition of rules for executing the production and use of data and data-related resources. Data management starts with the rules that define them. The purpose of the definitions is primarily to provide quality data models that meet organizational requirements. The definition of data quality affects how data is produced and has a direct impact on how data is used. Since data quality is an important aspect of the data management lifecycle, it is good to do it certain to execute and assert authority over how they are defined. This means that we have to govern the process of how we define the data. Again, data modeling is a discipline that needs to be governed, making data modeling a form of data governance. Data modeling is focused on the conceptual, logical, and physical definition of data. Each of these phases includes different levels of abstraction. Models very often give rise to databases and data resources that become part of the information system.

2.4.1 NGSI-LD

NGSI-LD ("Next Generation Service Interfaces - Light Data) is an information model that prescribes the structure of context information. The context must be based on the NGSI-LD information system. The model specifies the data representation mechanisms that will be used by the NGSI-LD API.

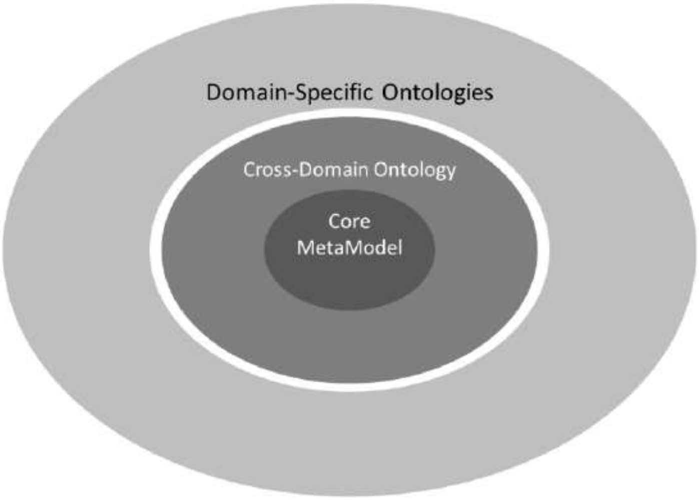


Figure 2.4: Overview of the NGSI-LD Information Model Structure.

The NGSI-LD Information Model is defined on two levels (Fig. 4.2.1-1): the foundation classes that correspond to the Core Meta-model and Cross-Domain Ontology. The first is equivalent to a formal specification of the "property". It is a set of generic and transversal classes that aim to avoid conflicts or redundant definitions of the same classes. Below these two levels, it is possible to devise domain-specific ontologies or vocabularies. NGSI-LD is very flexible. The standard definitions are well documented and allow representation of broad contexts. However, the big advantage of this standard is that you can define new properties to be used for specific purposes.

2.5 Big Data

Big Data are and will be the basis of new phenomena of exponential growth and innovation, capable, if well managed and interpreted, of increasing knowledge and productivity of many sectors, in particular the Public Administration, healthcare, finance, electronics, and information technology industry. In general, Big Data makes it possible to identify everything related to the analysis, processing, and storage of large amounts of data, which often originate from multiple sources. Big Data technologies are based on an interdisciplinary approach that takes into account mathematics, statistics, and computer science advances. Big Data is defined using the Laney paradigm in which we talk about the 5 Vs of Big Data (Fig. 2.5): volume, velocity, value, variety, and veracity.

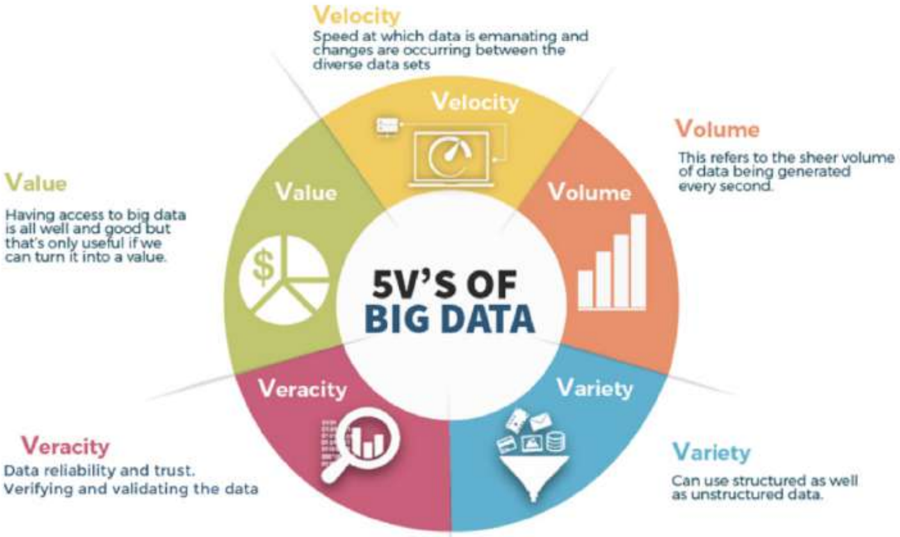

Velocity
Speed at which data is emanating and changes are occurring between the diverse data sets
Volume
This refers to the sheer volume of data being generated every second.
Value
Having access to big data is all well and good but that's only useful if we can turn it into a value.
Veracity
Data reliability and trust. Verifying and validating the data
Variety
Can use structured as well as unstructured data.
Velocity
Value
Volume
Veracity
Variety
5V'S OF BIG DATA

Figure 2.5: Big Data 5V.

2.5.1 Volume

The volume of data that Big Data-based solutions work on is substantial and always growing. The order of magnitude is billions of gigabytes with millions of servers working on their storage and processing. These volumes of data require different standards for their memorization and processing, as well as for the processes of preparation, care, and management. The main sources of large volumes of data are:

- Data collected by sensors;
- Statistics;
- Social Media;

- Results of scientific experiments.

2.5.2 Velocity

Data arrives quickly, and huge datasets can accumulate in small periods of time. This situation affects the processing time maximum for this data to be usable. It is necessary to design efficient and elastic data processing solutions, which take high data into account storage capacity. It should be considered that not all sources have the same speed and data range.

2.5.3 Value

Value refers to the utility that the data can have. The value of data is related to veracity: the greater the fidelity of the data, the greater its value. The concept of value Value is also strongly related to the length of data processing. The results of data analytics have an expiration date; "old" results have no value. It can therefore be deduced that the longer it takes to transform the data, the lower its value will be.

2.5.4 Variety

Data variety refers to the different formats and types supported by big data solutions. It is essential that Big Data management systems take into account the different formats that can arrive from sources and can still process them without interrupting execution.

2.5.5 Veracity

Veracity refers to the quality and fidelity of the data. Input data must arrive in "clean" and quality Big Data environments. Action is required to achieve this goal of data processing to remove invalid data and noise. The data will consist of signal and noise. Noise indicates that data cannot be converted to information and, therefore, has no value. The signal indicates data that can be translated into meaningful information. The higher the signal-to-noise ratio, the higher the veracity. Usually, if the data is acquired in a controlled way they have a lower noise level; this shows us how the noise also depends on the source of the data and not only on their type.

2.6 Blockchain Technology

The Blockchain is a new and innovative technology that is revolutionizing the world. It can be defined as an immutable ledger for recording transactions, maintained within a distributed network of mutually untrusting peers. Every peer maintains a copy of the ledger. The peers execute a consensus protocol to validate transactions, group them into blocks, and build a hash chain over the blocks (Fig. 2.6). There are many different types of Blockchains.

- **Public Blockchains:** are large distributed networks that are run through a native token. They're open for anyone to participate at any level and have open-source code that their community maintains.
- **Permissioned Blockchains:** control roles that individuals can play within the network. They're still large and distributed systems that use a native token. Their core code may or may not be open source.
- **Private Blockchains:** tend to be smaller and do not utilize a token. Their membership is closely controlled. These types of Blockchains are favored by consortium's that have trusted members and trade confidential information.

The Public Blockchain is the one that supports cryptocurrency and is permissionless, anyone can join it without a specific authorization. These Blockchains use the consensus protocol based on proof of work (PoW). Permissioned Blockchains provide the ability to protect interactions between a group of users that have a common goal but do not trust each other. The Permissioned Blockchain relies on the identity of his peers and by doing so uses the Byzantine-fault tolerant (BFT) consensus. The BFT protocol is based on the problem of the Byzantine General, which states that, in a group of people consensus is reached on the strategy to be adopted unless there is a traitor among them.

2.6.1 Consensus Protocols

One of the most important and innovative aspects of Blockchain technology is the consensus protocols. These, through a distributed network, can create a system of agreement between different irrefutable devices. This protocol allows us to keep all nodes on a network synchronized with each other. It is important that there is a consensus mechanism in the phase of insertion of information into the Blockchain. Anyone can store data on the Blockchain but with the consensus mechanism. The term consensus is used to indicate the agreement of the nodes on the network with the same status in the Blockchain.

Consensus Protocol Rules

A consensus algorithm can be defined as the mechanism by which a Blockchain network reaches consensus. Consensus algorithms are used in public (decentralized) Blockchains where nodes must agree on the validity of transitions. These algorithms ensure that the rules of the protocol are followed and that all transactions take place correctly. There are different types of consensus algorithms. The most common implementations are PoW and PoS.

Proof of Work (PoW)

PoW was the first consensus algorithm created and is an essential part of the mining process. Mining is the process by which transactions between users are verified and added to the public register of the Blockchain. Furthermore, is one of the key elements that enable cryptocurrencies to function as decentralized peer-to-peer networks, without the need for a central authority. A miner is a node within the network that collects transactions and works to organize them into blocks. When transactions are made, the miner nodes receive and verify them, adding them to the memory pool and starting to assemble them in a transaction block. The first phase of the mining process is to produce the hash of each transaction contained in the memory pool. Before starting, the miner node adds a transaction in which it sends itself the mining reward. This transaction is indicated with the transaction name "coinbase". It is the operation through which coins are created from 'nothing' and in most cases, it is the first transaction in a new block. Once generated, all the hashes are organized in a structure called Merkle Tree, or hash tree. Thus the hashes are matched in pairs producing the hash of the result, repeating the process until reaching the "top of the tree", also called root hash or Merkle root.

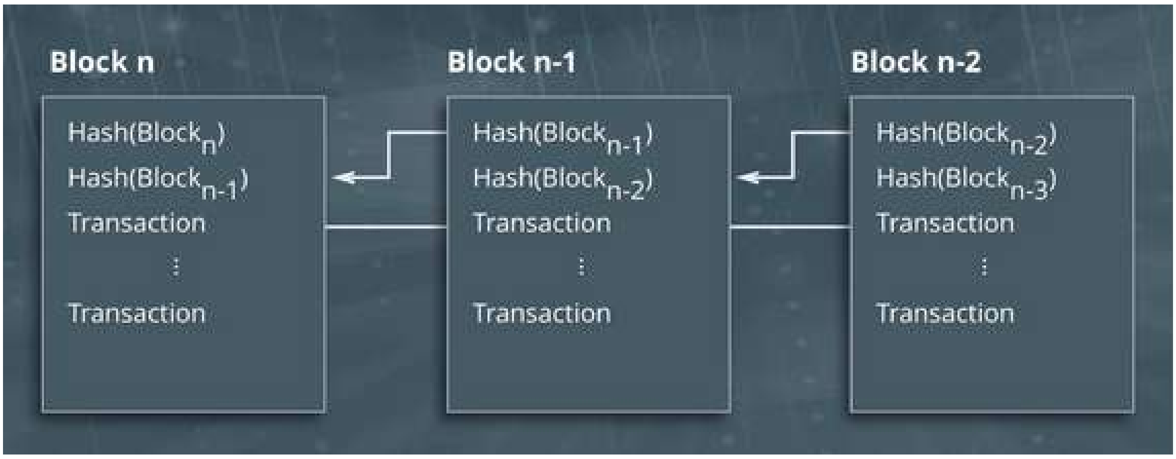


Figure 2.6: Blockchain Blocks Creation.

The hash root, along with the hash of the previous block and a random number called nonce, is inserted into the block header. Passing this header into a hash function produces an output that will act as the block identifier. Simply put, the Proof of Stake consensus algorithm replaces PoW mining with a mechanism where blocks are validated based on the participants' stakes. The validator of each block (also called forger or minter) is determined by an investment of the same cryptocurrency and not by the computing power used. Each PoS system can implement the algorithm in different ways but, in general, the Blockchain is protected by a pseudo-random election process that considers the richness of the node and the age of the coins (how long the coins have been frozen or frozen in the stake) - together with a randomization factor. Currently, the Ethereum Blockchain is based on a PoW algorithm, but the Casper protocol will eventually be released to pass the network from PoW to PoS to try to increase its scalability.

Proof of Stake (PoS)

This consensus method is based on the principle of putting something "at stake". In the PoS consensus algorithm, a new block is chosen in a deterministic manner, depending on the amount of cryptocurrency that the validators have. This means that block reward and mining are missing, as no cryptocurrency units are created with the creation of each block. Validators are compensated by earning a commission for validated transactions. There are different types of PoS Systems that depend from this different aspects:

- **Concept of Seniority:** the amount obtained through the product of the quantity of coins for the number of days in which these coins were frozen;
- **The speed of coin circulation:** as per ReddCoin;
- **Based on a vote:** where the block creators can be selected by DPoS (Delegates Proof-of-Stake) as per Lisk.

The PoS algorithm establishes that to undermine blocks it is necessary to possess and block more or less large amounts of cryptocurrency (defined as staking). Thanks to staking, the owner of the portfolio validates a block periodically, thus obtaining a constant percentage return. This algorithm is the solution to the problems we have with PoW. The system does not require a large computational power, it is more economical in energy costs. To compromise the network would need a majority of 51%, making the operation very expensive and complex. To make an attack it would be necessary to buy 51% of the cryptocurrency supply

and this would lead to a collapse of its value after the attack.
The PoS algorithm is considered better than the PoW. In addition, other methods are available, such as the **Delegated Proof-of-Stake**, where cryptocurrency owners can vote for a representative who can propose changes to the system and share mining rewards. PoS Blockchains have disadvantages. Thousands of transactions per second cannot be achieved for a network of payments.

2.6.2 Smart Contracts

A key element within the Blockchain is the transaction. These contain all the information necessary for the transfer of properties from one account to another. The Smart Contracts are used to describe the transfer of data of a property within the Blockchain. A smart contract is a machine-readable description of the will of the parties involved. Smart contracts can be defined as autonomous computer programs written in a specific Blockchain programming language. Unlike transactions, smart contracts are much more flexible as far as the subjects, objects, and parties involved in a transaction. A transaction can implement a smart contract or can perform a function on an existing one. It is possible to break down the process for creating this type of contract into 6 key steps:

1. Two or more parties identify a common interest;
2. Write a smart contract together, setting the desired conditions and effects;
3. The same Blockchain becomes the guarantor of the contract;
4. When consent is obtained in the network, the contract "executes" its conditions;
5. Insert the smart contract into the selected Blockchain;
6. After the conditions have been fulfilled, the Blockchain will be updated by the system status change;

The digital form typical of smart contracts is regulated both internationally and in Italy. Smart Contracts, "children" of code execution, however, present some limits that they are trying to overcome: the greater certainty, predictability and reliability of the smart contract is ensured at the expense of less flexibility and preserving the necessary margins interpretation of which the contract cannot do without.

2.6.3 Public Blockchain

Is important to know that a public Blockchain is a type of blockchain without authorization. Each user can access the Blockchain, read information, enter new ones, or make transactions. The main aspect of these Blockchains is being decentralized and secure, the data cannot be changed once inserted into them. Public Blockchains are open, everyone can write data to the Blockchain and read the information contained in it. The most popular Blockchain platforms are Bitcoin and Ethereum. These technologies have been created with the purpose of protecting anonymity. If the user is not known, there is no way to create authorizations, define accesses based on roles, and verify data that can be encrypted or read. Having anonymity is one of the main advantages of using cryptocurrencies, so it is more advantageous to use a public Blockchain.

Public Blockchains have disadvantages. Each block contains a record of many transactions on the network, but there may be many participants in the network itself. For this reason, the reward is kept low and that of the person who creates the next block is regulated. We use the Proof of Work algorithm, which manages to solve a problem and acquires the right to create a new block. The main problem is that these problems require a lot of computational power and resources. These Blockchains are suitable for public and non-business use cases.

2.6.4 Private Blockchain

For the protection of privacy and data security, companies set up and configure private Blockchains. They are Blockchains that do not allow access to anyone, to gain access you must have an invitation that is validated by the administrator or a set of predefined rules at the same time. These networks are known as authorized networks and there is a limitation on who is authorized to join. Private Blockchains have the ability to limit the activity of the participants so that transitions can be performed only by some authorized participants who can make transactions and not by others who are in the network but without authorization. This allows for an additional level of privacy. The rules for participating in the Blockchain can be established by existing participants by an authority that defines the policy or by the consortium. All network participants play a role in maintaining the decentralized Blockchain. Only entities participating in a given transaction may be aware of them, while other entities may not have access to them and information about them. These Blockchains are lighter and provide transactional throughput of a few orders of magnitude higher than public Blockchains. While public Blockchains have limited operations, private Blockchains - led by companies - have the power to revolutionize many aspects of everyday life.

2.6.5 Ethereum

Ethereum is a project that develops as a public Blockchain in the open-source distributed computing platform, created to make available the possibility of creating and managing Smart Contracts in peer-to-peer mode. Ethereum can be defined as a Distributed Computing platform, and one of the basic elements of this project is the EVM or the Ethereum Virtual Machine.

We can think of Ethereum as the largest shared computer capable of delivering a very high amount of power everywhere and forever. It is a platform that can be used by all those who want to join the network and also allows all users to have an immutable and shared archive of all the operations that are performed. Ethereum is designed to be adaptable to different systems and can be easily applied to create new applications. Ethereum is a Programmable Blockchain that does not limit itself to providing predefined and standardized "operations", but allows users to create their own "operations".

Ether

Users of an Ethereum Blockchain can interact on a peer-to-peer network, they can develop and interact with each other using the computational resources of the network. Using these resources comes at a price, and the cost is paid with a virtual currency called Ether. Ether is concretely a token that is treated as a cryptocurrency exchanges with the ticker symbol of ETC. Ethereum also has Gas, an Internal Transaction Pricing Mechanism. The purpose of the Gas is to optimize the resources of the network, and to allocate resources correctly and proportionately according to the required functions.

Ethereum Virtual Machine EVM

Ethereum Virtual Machine (EVM) is the Ethereum engine that is represented by a runtime environment for the development and management of Smart Contracts in Ethereum. The EVM operates in a completely safe way, as it is completely separate from the network. The code managed by the Virtual Machine does not have access to the network and the Smart Contracts themselves are independent of each other. Ethereum is a complete Turing system that allows developers to create applications that run on EVM, using programming languages that refer to languages such as JavaScript and Python.

Ethereum Contracts

The contracts in Ethereum are Smart Contract, which thanks to the remuneration assessed in Ether allows the management of contractual services in a secure and public way. Ethereum users can have the Ethereum Virtual Machine (EVM) able to execute algorithms on a global network based on the nodes of all participants. Each participating node of the network is compensated or can compensate in Ether. Ethereum allows developers to program their smart contracts, the so-called white papers. The language with which they are implemented is Turing-complete, which allows for a much wider set of computational instructions.

2.7 Artificial Intelligence

The term Artificial Intelligence (AI) refers to the ability of an automated system to perform actions commonly associated with entities with their intelligence. AI can be defined as a set of computer techniques applied for the simulation of human intelligence through optimized software. Despite continuous advances in data processing speed and memory capacity, there are still no AI systems that can match the flexibility of the human mind across different domains or in tasks that require daily knowledge refresh. On the other hand, some AI applications have achieved high performance comparable to that of industry experts or professionals in performing certain tasks. In the last period, AI has entered the mainstream through widespread familiarity with the "Generative Pre-Training Transformer" applications, of which the most popular and widespread application is OpenAI's ChatGPT, which has become, erroneously, synonymous with artificial intelligence. The main characteristic of AI is its ability to acquire information, rationalize, and take actions that have a given probability of achieving a specific goal. There are different forms of learning applied to AI, the simplest is to learn by trial and error while the implementation of generalization techniques turns out to be much more complex. Generalization involves applying experience to new analogous situations thus simulating reasoning. The process of reasoning is based on drawing appropriate conclusions according to certain situations. This definition is called inference. In general, inferences are classified as deductive or inductive:

- Deductive inferences: the truth of the premises guarantees the truth of the conclusion;
- Inductive inferences: the truth of the premises supports the conclusion without giving absolute certainty;

The Scientific community is divided into three AI schools of thought: weak AI (weak

AI), strong AI (strong AI), and Assistive AI. To give a practical example of this vision voice assistants (Google Assistant, Siri, etc) are considered weak AI applications. They operate within a limited set of predefined functions. On the other hand, self-intelligent machines capable of making decisions autonomously without human interference are considered strong AI systems.

Weak AI

It is a research approach based on advanced algorithms to perform simple problem-solving tasks that concern only a part of the sphere of human cognitive ability. Weak AI forecasts using models and is capable of transforming huge amounts of data into knowledge. However, a limitation of the weak AI is the potential damage that its choices can generate. In the event of situations such as autonomous driving, a wrong choice can cause serious problems for humans.

Strong AI

Strong AI refers to the school of thought according to which automated systems and/or machines can develop human consciousness. According to the principles of strong AI, machines can be equipped via software with their intelligence used to perform complex actions and acquire new capabilities independently.

Assistive AI

Assistive AI is based on the use of "enhanced" models of artificial intelligence. It aims to support man with complex tasks, enhancing thought processes and making them more efficient. An example of this technique is the X-ray dark spot analysis system. These are systems that analyze the products that leave the production lines, evaluating their quality.

2.7.1 Machine Learning and Deep Learning

In common parlance, the terms "Machine Learning" (ML) and "Deep Learning" (DL) are erroneously used as synonyms of Artificial Intelligence.

It is clear that there is a substantial difference between them. A representation of the terminology and the relationship between the terms used is schematized in Fig. 2.7. In this section we will describe the main characteristics of ML and DL.

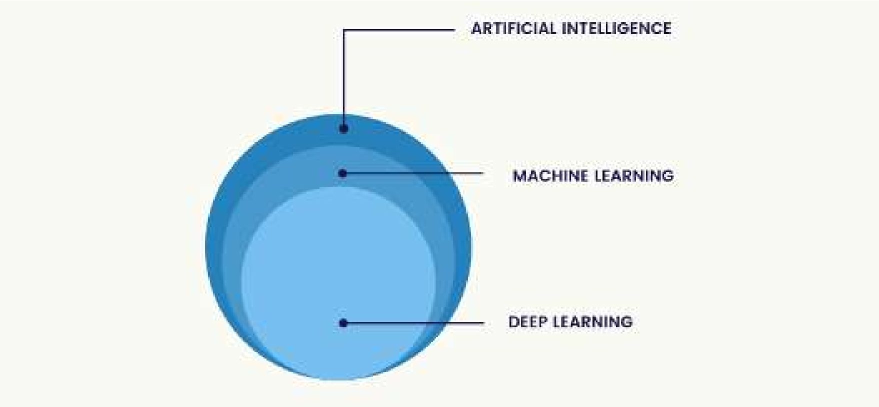


Figure 2.7: ML vs DL.

Machine Learning

Machine Learning refers to those techniques that deal with data management by simulating human learning. These techniques make use of complex algorithms optimized to increase the precision of the learning process and reduce errors in the different phases.

Machine learning can be divided into three main phases:

1. decision-making process: through ML algorithms, a prediction or classification is carried out on the dataset under analysis;
2. Error function: Evaluate the model's prediction. The models are compared to known data to verify their accuracy;
3. Model optimization: through optimization algorithms, we try to increase the accuracy of the built model up to a threshold value. The aim is to reduce the differences between the estimated model and the reference model.

The algorithm training process is of fundamental importance. It involves using statistical models to classify data which then allow predictions to be made. The information deriving from ML processes serves to guide the decision-making process within AI applications used in Public Administrations or companies.

Deep Learning

Deep Learning is a subset of Machine Learning defined as a neural network with three or more levels. Neural networks are non-linear structures that simulate the behavior of the human brain working on Big Data. Many AI applications are based on deep learning techniques in order to improve automation processes without requiring human intervention.

The type of data processed and the learning methods make it possible to differentiate deep learning from machine learning. While in ML structured data is used to make predictions, in DL the preprocessing part of the data is eliminated and they are used unstructured. This difference makes it possible to do without human intervention.

2.7.2 Machine Learning Models

Artificial Intelligence and everything that revolves around it deserves a broad discussion. However, in order to understand the following chapters, it is also useful to mention machine learning models. Machine learning models can be classified into three categories:

- Supervised Learning: Supervised learning requires labeled input and output data during the training phase of the machine learning model. Training data is labeled in the preparation phase before it is used to train and test the model. Once the model has learned the relationship between input and output data, it can be used to classify new datasets and predict outcomes. This technique is used to classify "invisible data" into established categories and future trends as a predictive model.
- Unsupervised Learning: Unsupervised machine learning is the training of models on unlabeled training data. This method is applied to identify patterns and trends in raw datasets, or similar data in clusters. In unsupervised machine learning, data is grouped by similar characteristics or by analyzing datasets for underlying patterns. These features make this model a powerful tool for gaining insights from data.
- Reinforcement Learning: Reinforcement learning (RL) is a subset of machine learning that allows an AI system to learn through trial and error. This technique uses feedback from your actions which can be positive or negative. The goal of the RL is to find the most suitable action pattern to maximize the total cumulative reward for the RL agent. It is interesting to observe that without training datasets, the limits of the RL are solved by the same actions of the agent with the input coming from the external environment. There are three types of RL:
    - RL based on policy or deterministic strategies that maximize cumulative reward;
    - RL based on Values based on maximizing an arbitrary value function;
    - Model-based RL allows you to create a virtual model for a given environment where the agent learns to operate within given constraints;

CHAPTER 3

Big Data Management in Urban Mobility

The urban transformation of the last few decades has introduced profound changes in key sectors such as urban mobility. The introduction of disruptive ICT technologies has made it possible to modernize several aspects of urban mobility by introducing what is now called Smart Mobility. Urban mobility and smart mobility contexts, but not only, now require more than ever the use of large amounts of historical data to carry out the necessary analyses for different use cases. A good management of time series data, able to use pagination concepts in an optimized way and providing the user with specifications functions, therefore become indispensable. The European Commission has funded many initiatives relating to urban transformation with particular attention to smart mobility. URBANITE (Supporting the decision-making in URBAN transformation with the use of dIsruptive TEchnologies) is an H2020 project investigating the impact, trust, and attitudes of civil servants, citizens, and other stakeholders concerning the introduction and adoption of disruptive technologies (e.g. AI, Decision Support Systems, big data analytics) in decision-making processes related to the planning and management of urban mobility. The project experiments, and validates its approaches and tools in the context of four real use cases in the cities of Amsterdam (NL), Bilbao (ES), Helsinki (FI), and Messina (IT). This chapter summarises the main findings matured during the first half of the project in the four cities, their main mobility issues and how disruptive technologies can play a role in supporting the decision-making process to solve them. Despite the four cities face different kinds of mobility issues and are characterised by different levels of IT maturity, we identified a chain of three categories of technologies that can improve the efficiency and effectiveness of decision-making processes in all four cities: data access and harmonisation, data analysis and data visualisation. This chapter summarizes the issues encountered in addressing mobility issues and how disruptive technologies can play a role in supporting the decision-making process to solve them. In particular, the problems concerning the organization of data will be analyzed in order to make querying the databases used by decision support applications efficient.

3.1 Optimized Data Management for Smart City Policy Decision

Nowadays, nearly all cities are increasingly moving towards models of intelligent infrastructures, the so-called smart cities, capable of anticipating the needs of their inhabitants by providing them with increasingly innovative and targeted services. A smart city can collect an infinite amount of information by acquiring real-time vehicle locations, weather data, air quality measurements or GPS position of electric vehicles in the city center. By processing this information it is possible to provide both administrators and citizens with real-time information or digital services that can improve their quality of life. The large amount of data exchanged between IoT acquisition tools (such as sensors and cameras) and databases, but also the need to make this information immediately available, makes central the creation of increasingly efficient data models, both in the process of writing data into the selected database and in the reading phase, when the access time by users becomes extremely important to increase the quality of the service. The correct data management is in fact important in various aspects of smart cities. The use of data is fundamental for the monitoring of cities and for the creation of planning and decision making tools. The European Commission funds various research programs for the development of smart cities and for an optimal use of data. With Horizon 2020, Horizon Europe and the Next Generation EU programs are being developed projects in various fields including that of urban mobility. The URBANITE project is active in this particular sub-area of Smart Cities and it was funded under the H2020 funding program. Among the objectives of URBANITE, the main one is to promote the use of disruptive technologies in the nascent Smart City in technological term, through the use and analysis of Big Data, artificial intelligence algorithms, etc. An innovative element, however, is that linked to the promotion of innovative tools for participation in decision-making processes such as the Laboratory Social Policy (SoPoLab). The aim of the project is therefore to provide stakeholders of the project a series of innovative technological tools in order to support the decision-making processes of the executives of public administrations and companies. At the base of these tools is important the organization and management of data, the subject of the work presented.

In the use case of the project, concerning the city of Messina, we faced the problem of organizing large quantities of data concerning local public transport. We have tried to propose an efficient solution for querying a huge database by introducing innovative concepts and using existing tools effectively for our purpose. The study focused on the optimization of data acquisition and their query in the case of using MongoDB. We implemented and tested the solution trying to understand if it could have a real beneficial effect in the implementation of decision making tools in urban mobility applications within the project.

3.1.1 State of the Art

In mobility use cases, sensor data or machine-to-machine communications are stored, analyzed and tracked using time series. However, in cases where systems need to be highly scalable and different functionalities are required, it could be better to use a document database such as MongoDB. In this study, therefore, we want to understand how the problem of querying large databases, based on time series, can be optimized using MongoDB, but in the first instance we want to investigate the state of the art to better understand which are the other solutions available. A general study of traditional relational databases as well as NoSQL-based solutions for time series data is reported in [25]. The authors conclude that for time series databases the key factor is the speed. The study shows that NoSQL databases can be used in case of time series which an high frequency of measurements. The solution indicated in this case, without considering MongoDB in the study, is Cassandra. An example of a DB comparison that also includes MongoDB is shown in [26]. In this paper the authors compare relational databases such as Oracle with non-relational DBs such as MongoDB, Redis and Cassandra. For the experiments a DB on railway connections is used. The comparison between the DBs is made with two models. The first concerns the speed of access to data on a small portion of the database. The second, on the other hand, evaluates the times and requests for storage space on a large number of records. The study carried out concludes that certainly on a large amount of data it is better to use NoSQL type DBs. In particular, MongoDB is the best in terms of query performance. In [27] it is experimentally demonstrated how to manage a time database, Chronos. The authors use temporal and parallel algorithms as well as specific RAM storage methods for data management. The presented method increases the efficiency of temporal data management by approximately 40% – 90% compared to other databases, such as MySQL and MongoDB. The increasingly common use of data for analysis purposes concerns various areas. Surely, an inexhaustible source of data are social networks. In this context a study, whose methodologies are interesting, is reported in [28]. After defining the application requirements, the authors make a detailed comparison of five of the most popular NoSQL systems, namely Redis, Cassandra, MongoDB, Couchbase and Neo4j, in relation to the defined requirements. The importance of defining the requirements before choosing a DBMS is fundamental. One of the problems in managing data organized in time series mainly concerns historical data. Especially in the field of mobility, various data accumulated over time are organized in SQL like databases. In [29] the authors tackle this problem by proposing a solution for converting queries from MySQL to MongoDB taking into account the database structure. In the IoT field and in particular in the context of mobility, the management of Big Data is fundamental. It emerges from several studies that the number of connected devices is always increasing, and the data they collect is managed more and more often with DBMS NoSQL. In practice, the data collected by IT devices are nothing more than series of data. This implies that it is necessary to manage the data with specific technologies. In [30] the authors present the results of an empirical comparison of three NoSQL Database Management Systems. The assessments cover Cassandra, MongoDB and InfluxDB, maintaining and recovering gigabytes of real IIoT data. The results of the tests show that MongoDB gave the best performance for queries on non-temporal indexed attributes, while Cassandra was unstable compared to its competitors. InfluxDB was on average the best performing solution, compared to Cassandra and MongoDB in terms of storage and with regard to ingestion and time-based queries. In [31] the authors evaluate the use of time series databases for telemetry data and they combine these results with microbenchmarks to determine the best compression techniques and storage data structures for designing a new optimized solution for data from IoT. The query translation method allows to use data models such as the Resource Description Framework (RDF) for interoperability and data integration in addition to optimized storage. The authors propose a framework, TritanDB, which shows an improvement in performance on cloud hardware on many databases used within IoT scenarios. In [32] the authors consider a case study in which IoT devices send large amounts of data to the database. In the context of this scenario, the performance of multiple DBMSs is analyzed. The results of the study allow to evaluate the load on the system that writes the data and the scalability of the system. From the evaluation of the results it is clear that MongoDB is the best choice, but according to specific configurations or needs, other choices such as MariaDB or InfluxDB are also optimal. The work presented in [33] reports a concept of a GPU extended non-relational database management system. The research focuses on implementing kernels and performing basic aggregate functions on a JSON file. The Numpy library, a CPU counterpart and MongoDB are compared showing the importance of the concept. The hypothesis is to test whether the GPU can speed up NoSQL database queries. The results show that GPU runtime grows steadily, but slowly. Furthermore, it was found that even in the case of 700,000,000 rows of integers the worst GPU time is 30.07 milliseconds, which is considered to be very fast compared to the CPU. However, it is possible to improve the performance of a DBMS containing Big Data. This strategy allows to optimize querying on large databases. A study on the use of this technique is reported in [34]. In [35] the authors propose SmartBench, a benchmark focused on queries resulting from (almost) real-time applications and long-term analysis of IoT data. The paper presents the evaluation of seven representative database systems and highlights some interesting findings to consider when deciding which database technologies to use with different types of IoT query workloads. The work highlights that the choice of the data base is strongly linked to the type of input, the data organization model and the type of querying that is planned to perform on the data.

Starting from the studies carried out, we propose in this section an optimization of the use of MongoDB in the management of time series data. In particular, given the innovations introduced by MongoDB 5.0 with respect to the management of time series, we will propose a solution that optimizes as much as possible the use of MongoDB and we will compare the results obtained to better understand which is more appropriate in our context.

3.1.2 Motivation

The URBANITE project was created to offer innovative technological solutions in the field of urban mobility. In particular, the aim of the project is to provide the pilot cities (Amsterdam, Helsinki, Bilbao and Messina) with innovative tools that allow the decision maker to make decisions and make assessments on mobility decision-making policies. These tools are based on data analysis and artificial intelligence algorithms and work on both historical data series and data collected in real time. The basic idea is that, starting from the analysis of historical data, it is possible to analyze the data acquired in real time, giving suggestions to decision makers. Therefore, the organization of the collected data and their structure is fundamental. In general, the data relating to this technological field are series of historical measurements acquired by the sensors, hence the reference to time series data. It is important to obtain quick answers from the analysis and above all, in case of open data policies, also to optimize data sharing efficiently. Several studies on the use of SQL and No-SQL databases have emerged from the state of the art. It is clear that the use of one technology rather than another also depends on the type of application to be implemented. In our case, however, being aware of the fact that MongoDB has released in version 5.0 the optimization functions for the management of time series, we want to deepen by pushing our approach as much as possible and comparing the results with the new solution. In particular, we want to use the bucket structure for managing time series data on document-oriented databases. After having structured the data in an optimized way for the software applications to be used in the field of mobility, we will compare the proposed solution with the approach used in MongoDB 5.0 for the management of time series. Starting from the opportunities offered by the URBANITE project, we worked on a database made available by the Municipality of Messina regarding the positions of local public transport vehicles. Within the project activities it was necessary to move data from an SQL database to a NoSQL, and from here we want to understand how to optimize the use of the database and what is the best approach for our purposes. The proposed work goes on by exploring complex approaches to data organization.

Furthermore, specific experimental approaches aimed at testing the functionalities of interest have been studied, precisely in order to evaluate in detail the computational differences of the applied approaches.

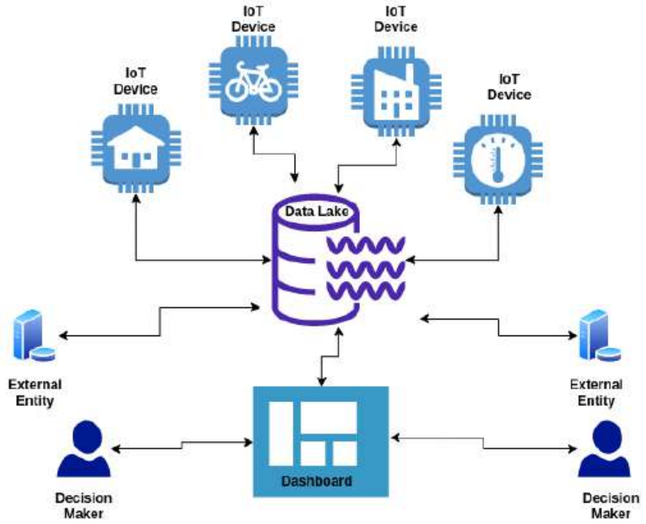


Figure 3.1: Generic use case, where sensor data are collected into a Data Lake and made available for use and sharing.

The proposed solution can be used in the context of urban mobility, but it has a general value. A generic use case is shown in Fig. 10.4, where several sensors in an environment record data and send it to a collection point, a Data Lake. On the Data Lake, the data would be organized according to the proposed solution and therefore will be easily usable by the

Decision Makers through a Dashboard, or shared with External Entities.

3.1.3 Methodology

In this section the methodologies implemented for the two approaches compared in this work are described. Our goal is to build an interface that allows a user to query large amount of time series data, considering different response formats and especially using a server-side managed pagination. Data can be returned as an array of documents, grouped by their timestamp or by their id.

The user specify a *date range*, a *pagesize*, *filters* (optional) and the *page id* through a RESTful API and get the data for that page as a response. In addition, data aggregation capabilities are offered, metadata properties can be enabled and included, and the total item count for the entire query is calculated (considering the number of results from all pages).

Time series collection

The new time series collection introduced in MongoDB 5.0 allows to easily handle time series data, making the data structure optimization transparent to the user.

First of all the collection has to be created into the database specifying the option required according to the measurements that are taking into account. Specifically, *timeField* is the name of the field that indicates the timestamp of the data, *metaField* is the field that contains the properties which are constant and the *granularity* is set according to the frequency of data collection. The data can be thus inserted into the database as if it were a standard collection.

The time series collections, for optimization reasons, have different limitations compared to the others. In particular, among other things, they do not allow updating or deleting documents, even if they provide the possibility of setting a TTL (time to live).

The pagination is implemented using the *skip-limit* technique. Therefore, considering the user's request, the page size is multiplied by the page id and the result gives the number of documents to skip. Instead, the query limit is directly set equal to the page size. As example, with a page size of 20,000 the page 0 will be obtained with a skip equal to 0, the page 1 with a skip equal to 20,000, the page 2 with a skip of 40,000 and so on.

The pipeline stages of the query are then dynamically defined according to a configuration file, which describes structure and characteristics of the collection. In summary, they retrieve the documents related to the specific page and organize them following the format chose by the user.

If requested, aggregation operation are performed in an added stage according to the operations (i.e. a subset of minimum, maximum, average) and to the granularity (seconds, minutes, hours, days, months, years) indicated by the user. It is important to say that in this case the pagination is more complex, because data that has to be aggregated together must be in the same page.

Finally, the total count is calculated considering the entire date range using the aggregation operator *$count* provided by Mongo. Since this value does not change passing from one page to another, it is calculated only at the moment of the first request and stored in cache for the following ones.

Advanced bucketing

This approach exploits as much as possible the use of buckets, using methodologies in order to optimize performances and functionalities to the maximum. It is based on five main pillars:

1. Bucket
2. Total count
3. Aggregation
4. Range pagination
5. Query pipeline

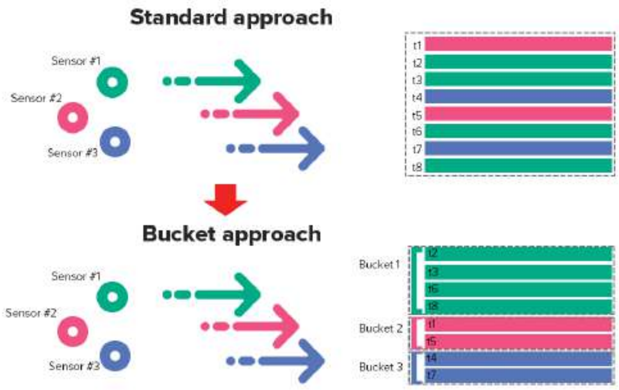


Figure 3.2: Standard vs Bucket approach.

Bucket

the **bucket** is a well known pattern that is used to optimize the management of time series data. In general, it is a group of documents, based on a specific expression and boundaries. If we consider periodic measurements that then provide time series data, the simplest and most spontaneous way to organize such data is to create a document in the collection for each individual measurement. Here, with the buckets approach, these documents are instead grouped with predetermined criteria into container documents (Fig. 3.2). Consequently, the collection contains buckets that consists of an array of documents, which represent the real measures, as well as contain the metadata that remains constant over time.

The bucket length can be fixed if there is set a maximum number of measurements, dynamic if the insertion of a measurement within it depends on other factors. It is also important to consider that the length of the bucket must take into account the limits that MongoDB imposes on BSON documents, which is 16 megabytes.

This work aims to provide first of all clear guidelines about the bucket structure that can be used in our contexts and can ensure optimal performance on all the features previously described. The most convenient bucket structure for our purposes is a dynamic one, in which a single bucket contains the measurements coming from a specific data source in a given time interval, because this will allow us to use specific algorithms for data retrieval. The criteria of insertion are thus based on both the data source and the timestamp of acquisition.

In summary we can describe the structure of our bucket composed of the following fields:

- an **_id**, the document unique index create automatically by Mongo;
- an **id**, which uniquely identify the data source (for example, a device-id);
- a **timestamp**, which sets the start of the time interval;
- a **granularity**, which establishes the size of the time interval;
- a list of **properties** (optional), that are the metadata related with the measurement;
- a **count**, which keeps track of the number of measurements contained in the bucket;
- an list **data**, that contains a document for each measurements.

It is worth noting that *id* and *timestamp* together represent the primary key. The granularity must be set according to the data acquisition frequency, in order to be sure to remain within the size limits of the single document (usually a reasonable size of a bucket can be around 1,000 measurements).

With the bucket approach, the insertion of a new measurement consists in the creation of the bucket if the document does not exists, otherwise it will be done through an update operation on the reference bucket that adds the new measurements to the list of the bucket (in this case, to simplify the process, on MongoDB the update method with the upsert option can be used in both cases).

A generic structure of our bucket implementation is described below in Listing 3.1.

Listing 3.1: Bucket data structure example

```
{
  "_id": ObjectId("60b0c77517fe39f60ed63e45"),
  "id": "sensor01",
  "timestamp": "2021-01-01T00:00:00Z",
  "property1": "generic",
  "property2": "description",
  "granularity": day,
  "counts": 55,
  "data": [
       {
            "timestamp": "2021-01-01T01:00:00Z",
            "value1": 25,
            "value2": 5,
              ...
       },
       {
            "timestamp": "2021-01-01T01:01:00Z",
            "value1": 6,
            "value2": 32
       },
          ...
    ]
}
```

Total Count

the second pillar concerns the calculation of the **total count** of the items corresponding to the request. In our solution we want to find a better approach for counting to the one offered by MongoDB, which is quite slow if the amount of data is significant. For this reason we build a special data structure that we store in a cache and that allows to calculate the value quickly, regardless of the amount of measurements accounted.

Considering all the buckets in the collection sorted by timestamp, for each of them we build a parallel structure that calculate the cumulative count values with respect to the previous bucket. To give a simple example, let's consider the following bucket counts:

- Bucket A count = 10;
- Bucket B count = 15;
- Bucket C count = 20;
- Bucket D count = 5;

The parallel structure will be then dynamically built as follows:

- Bucket A cumulativeCount = 10;
- Bucket B cumulativeCount = 10 + 15 = 25;
- Bucket C cumulativeCount = 25 + 20 = 45;
- Bucket D cumulativeCount = 45 + 5 = 50.

Since the counts are cumulated and the buckets are ordered by timestamp, to know the number of elements contained in the request interval it is sufficient to simply know the *cumulativeCount* values of the first and last bucket of the request, regardless of everything in between.

Considering the previously described example and a request whose timestamp match includes buckets from B to D, the totalCount can be calculated with the following formula:

$$cumulativeCount\_D - cumulativeCount\_B + count\_B$$

which is equal to $50 - 25 + 15 = 40$ and is equivalent to the sum of the single bucket counts. As we can see in the calculation, we are completely ignoring the buckets between B and D, in this case C.

Actually, the necessary approach must take into account two additional aspects:

- the time interval of the request may exclude some measurement of the bucket delimiters, when specifying a higher granularity;

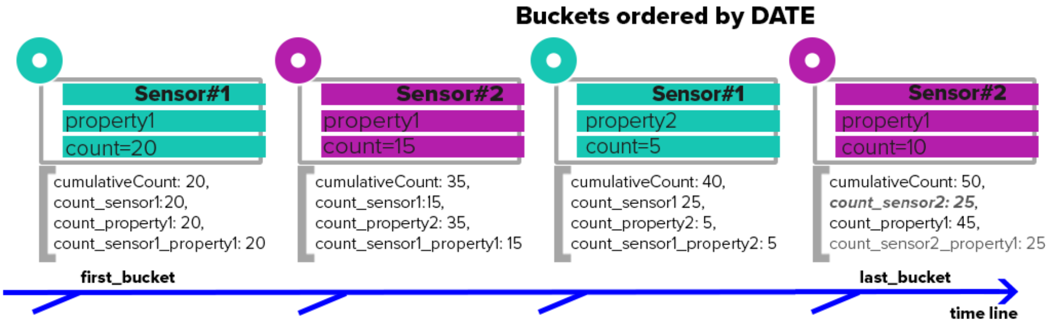


Figure 3.3: Buckets parallel structure with cumulative counts (buckets are sorted by timestamp.)

- some buckets can be excluded because the user can specify filters on the id and metadata, where permitted.

For this reason the algorithm that calculates the total count foresees the following steps:

- the parallel structure, in addition to the *cumulativeCounts* described above, calculates also the cumulative values taking into account the possible filters, as in the example shown in Fig. 3.3;
- considering the request, the bucket delimiters are detected;
- using the parallel structure and the formula previously described, the *totalCount* is calculated;
- a query checks how many measurements are inside the delimiters but outside the time range of the request. This value is then subtracted from the *totalCount* (this operation is quite fast because it operates on the few delimiting buckets).

Once the *totalCount* is calculate, the value is stored in a cache associating it to the request via hash code. Since the parallel structure is independent from the requests, it is calculated a priori and stored in-memory to be used when required. When a new measurements is inserted, the parallel structure has to be updated.

Aggregation

the third pillar, the **Aggregation**, is closely linked to the concepts of *granularity*. First of all in our bucket structure we set a field that contains pre-aggregated data with a granularity equals to that of the bucket. This means that, for each new insertion of a measurement within the bucket, the pre-aggregated values of the bucket are updated in terms of minimum, maximum and average. The final generic bucket structure is shown in Listing 3.2.

Listing 3.2: Bucket data structure example with pre-aggregation

```
{
  "_id": ObjectId("60b0c77517fe39f60ed63e45"),
  "id": "sensor01",
  "timestamp": "2021-01-01T00:00:00Z",
  "property1": "generic",
  "property2": "description",
  "granularity": day,
  "counts": 55,
  "aggregation": {
      "min": { "value1": 6, "value2": 5, ...}
      "max": { "value1": 45, "value2": 55, ...}
      "avg": { "value1": 23.4, "value2": 18.1, ...}
  }
  "data": [
       {
            "timestamp": "2021-01-01T01:00:00Z",
            "value1": 25,
            "value2": 5,
              ...
       },
       {
            "timestamp": "2021-01-01T01:01:00Z",
            "value1": 6,
            "value2": 32
       },
          ...
    ]
 }
```

At this point, depending on both the level of aggregation granularity required by the user and the granularity of the bucket itself, we can fall into three different cases:

1. granularity_aggr $<$ granularity_buckets
2. granularity_aggr $=$ granularity_buckets
3. granularity_aggr $>$ granularity_buckets

In the first case data is simply aggregated from collected measurements; in the second case we directly use the pre-aggregated data of the bucket; in the third case we aggregate the pre-aggregated data of the bucket (in particular for the average we consider a weighted sum based on the count value of the single bucket).

Range Pagination

the fourth pillar regards the **Range pagination**, which is an optimized alternative of the skip-limit approach for implementing pagination. In particular, it considers the documents of the collection (i.e. the buckets) in their natural order and uses match filters based on the document _id to select the measurements that has to be returned for the specific page.

To be more clear, when a user requests a page-(i), the _ids of the documents that will delimit the page are identified based on the *count* value of the individual buckets. These delimiters are calculated using an iterative approach: starting from a *cumulativeCount* of zero, the count of the next bucket that matches the filters and has an _id greater than those on the page-(i-1) is added, until this value does not exceed the *pagesize* indicated by the user. The first and last _ids will be then our page delimiters. In this case the size of the page depends on the bucket counts, therefore the parameter specified by the user in the request is more properly a *maxPageSize*.

Obviously, since to calculate page-(i) it is necessary to start from page-(i-1), this method performs best when pages are requested consecutively starting from page-0. Moreover, to avoid having to calculate the previous pages at every request, the values of the delimiters and the total count are stored in a special cache, in which each request is identified by its hash code.
An example of the cache structure is shown in the Listing 3.3.

Listing 3.3: Pagination cache example

```
{
 "requests": {
   "D8454A6029D6B9AED2468B1B411F5DBA": {
     "totalCount": 17000234,
     "pages": [
       {
         "id": 0,
         "$gte": ObjectId("60b0c77517fe39f60ed63e48"),
         "$lte": ObjectId("60b0c77517fe39f60ed63e89")
       },
       {
         "id": 1,
         "$gte": ObjectId("60b0c77517fe39f60ed63ea0"),
         "$lte": ObjectId("60b0c77517fe39f60ed63eac")
       },
       ...
     ]
   }
 }
}
```

In the third case of aggregation, when the required granularity is greater than the bucket granularity, great care must be taken with pagination since the data to be aggregated together must necessarily be contained in a single page. Consequently, in this situation, the page delimiters are forced to the extremes of the aggregation period according to the granularity.

Pipeline

the **pipeline** stages of the query are dinamically defined according to the request and a configuration file, which describes structure and characteristics of the collection. Data are organized depending to the format chosen by the user: by default the data are not grouped but are presented as a list of documents; alternatively, it is possible to group them using timestamp or id as a key. The results are then put together and returned in response to the user.

3.1.4 Experiments and Results

In this section the experiments performed to validate, test and compare both *Time series collection* and *Advanced Bucketing* approaches are presented. We will use a case study addressed in the URBANITE project in the context of smart mobility, in particular exploiting a real dataset containing the measurements of public transport extracted from an OpenGTS (Open GPS Tracking System) portal.

The measurements are collected at a distance of about ten seconds from each other when a vehicle is switched on. The time period considered is four years and provides a significant amount of data equal to approximately 178 million measurements. The information present concerns speed, altitude, status and GPS position of the reference vehicle.

The tests are carried out on two VMs having the following characteristics:

- VM-MongoDB:
  - Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz x4

  - RAM: 16GB
  - Disk Space: 197 GB

- VM-Server:
  - Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz x2
  - RAM: 4 GB
  - Disk Space: 20 GB

Data import

First of all the two type of collection are created and the measurements, taken from a MySQL database, are inserted one by one. For the Time series collection the granularity of measurements is set to *seconds*, while in the Advanced Bucketing the granularity of the bucket is fixed to *hour*. In the experiment 2000 measurements collected from the same device and in a interval of one hour are considered.

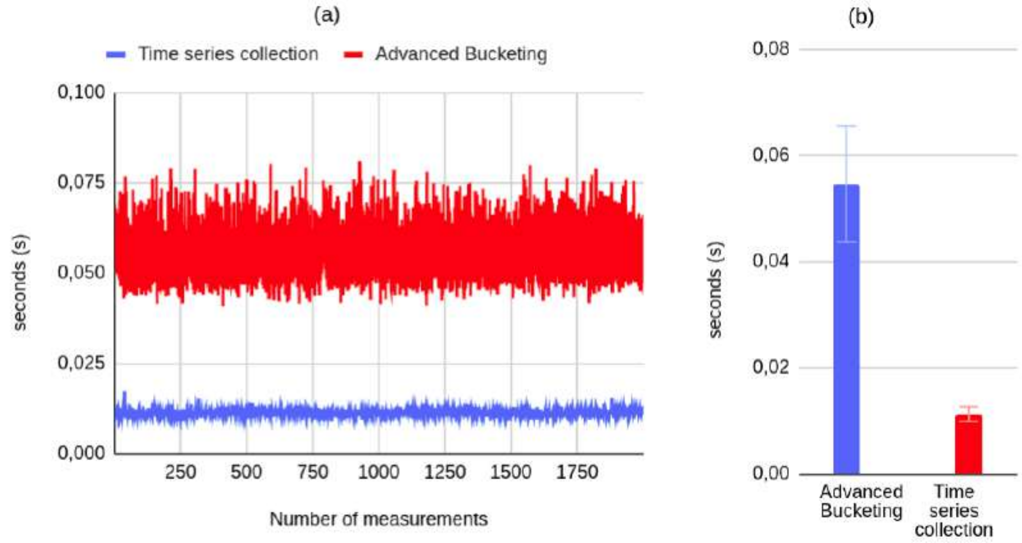


Figure 3.4: Comparison of data insertion times. (a) Time of insertion of 2,000 measurements (b) Average time of insertion.

As we can see in Fig. 3.4, the insertion times in both approaches remain more or less constant as the number of measures inserted increases. However, the average speed of the insertion time required in the Time series collection is almost five times less than Advanced Bucketing (Time series collection: 0.0546s - Advanced Bucketing: 0.0113s). This is closely related to the two different methods used, *insert* in the first case and *update* in the second, and to the pre-aggregation calculation performed in the latter. In order to speed up the process for the insertion into the bucket, an index on both *timestamp* and *data.timestamp* is created.

Also in terms of storage, as shown in Table 3.1, the first approach succeeds in being more optimized, even if at the level of dump size Advanced Bucketing clearly prevails.

Table 3.1: Bucketing Approach Storage Comparison

	Advanced Bucketing	Time series collection	% diff
bucketsCount	1541081	1388374	9,9
avgBucketSize(bytes)	17791	15744	11,5
storageSize(bytes)	6238638080	4579606528	26,6
dumpSize(bytes)	3533452435	5228679763	-47,9

Pagination

Pagination is the focal part of this work, as it is the main key needed to achieve adequate data retrieval and sharing, even when considering large amounts of data.

The tests are performed using different pagesize and requesting 1000 pages consecutively. As we can see in Fig. 3.5(a), in the case of Time series collection the behaviour is linear and the required time is accumulated following the increment of the request page number. This is due to the skip-limit approach, as page after page the elements to be skipped become more and more.

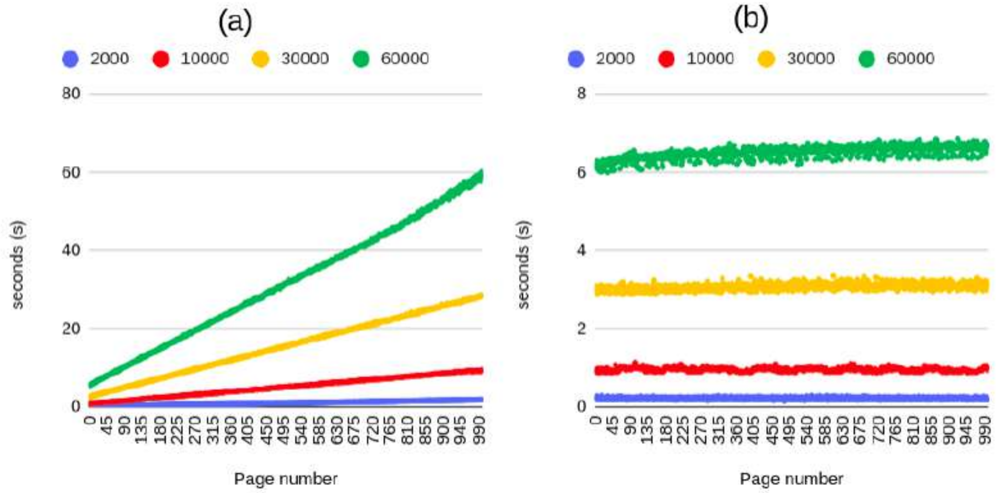


Figure 3.5: Comparison of data pagination times considering different pagesize. (a) Time series collection. (b) Advanced Bucketing.

The Advanced Bucketing, on the other hand, thanks to the Range pagination technique, manages to remain constant on a very small scale compared to the opponent's one, as shown in Fig. 3.5(b). Taking in consideration page-(1000), the time of calculation demanded in this case is 10 times less then that one of the Time series collection and this value grows more and more with the increase of the number of page. The trend in both cases remains unchanged as the pagesize value changes.

Data retrieval

Since increasing the size of the pages obviously also increases the time required for the single request, it becomes legitimate to ask what is the optimal size for obtaining large amounts of data and what is the difference between the two approaches. In this test we evaluate the time required for retrieval of 15,000,000 measurements, varying the pagesize as shown in the Fig. 3.6.

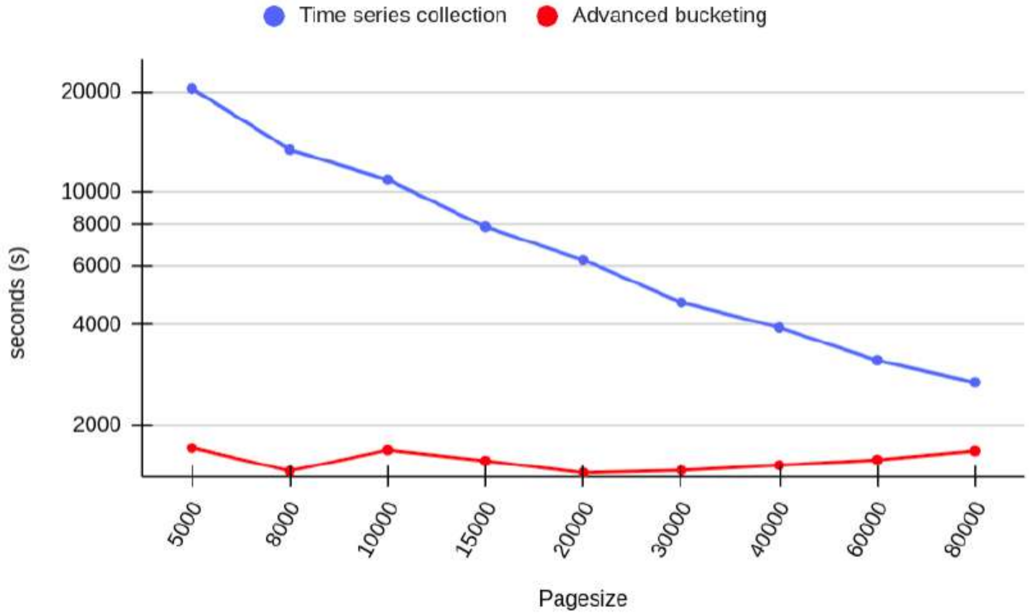


Figure 3.6: Comparison of data retrieval: Time Series vs Advanced Bucketing.

We can immediately see that in the case of Time series collection approach, the time required reduces as the page size increases. This is an expected behaviour and we can justify it because more pages correspond to fewer requests and, consequently, the same measures will have to be skipped fewer times. Ideally in fact, with a *pagesize* equal to *totalCount*, the results would be obtained with a single request and the overhead introduced by pagination would be reduced to zero. This is obviously not feasible in practice, since the size of the page must be contained, both for the processing capacity required and for network reasons.

On the contrary, according to the tests carried out on pagination, Advanced Bucketing allows us to have a more or less constant behavior as the pagesize varies. In this case, although the differences are minimal, the best performances are obtained with pagesize of 8,000 and 20,000 elements; we can attribute this behaviour to the size of the single buckets that, not being constant, has its influence on pagination.

In all page sizes tested, despite the fact that the two methods tend to converge, Advanced Bucketing consistently performs better than the Time series collection approach. Page sizes larger than those used are not considered appropriate for use in the reference context.

Response formats

Another relevant aspect to consider, in order to make easier for users the use of data, is the format in which the data is presented. In this case, as announced, by default the measurements are returned as a list of documents, as in the example shown in Listing 3.4.

Listing 3.4: Default results format example

```
{
    "results": [
        {
            "id": "784",
            "timestamp": "2016-01-02T12:00:23+00:00",
            "status": "EnRoute",
            "location": {
                "type": "Point",
                "coordinates": [
                    15.552624980919063,
                    38.2070849952288
                ]
            },
            "speed": 19.0,
            "altitude": 75.0
        },
        ...
    ]
    "status": 200,
    "startDatetime": "2015-01-01T00:00:00",
    "endDatetime": "2021-01-01T00:00:00",
    "pageCount": 9990,
    "cumulativePagesCount": 49660,
    "totalCount": 178539023,
    "page": 4,
    "nextPage": true,
    "lastUpdate": "2021-12-09T20:03:05"
}
```

Alternatively, the implemented algorithms allow the same data to be grouped using timestamps or ids. In Fig. 3.7 we can analyze the differences in terms of the computation time required to derive the data in the different formats. In the case of default (ungrouped) and data grouped by Id, we can see that the Time series collection approach performs slightly better than Advanced Bucketing. This is mainly due to the simplified data structure, which avoids small reorganization operations. In the case of grouping by timestamps, however, the use of buckets helps the process and causes Advanced Bucketing to perform better.

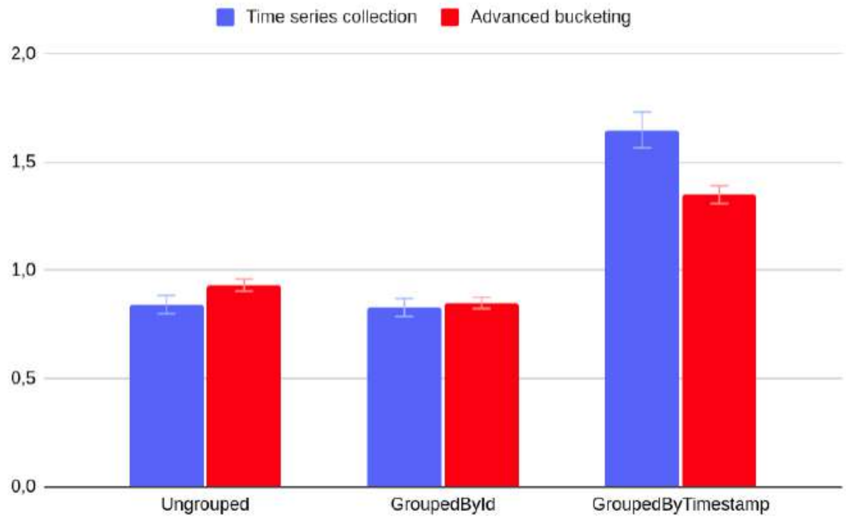


Figure 3.7: Comparison computation time between different data formats in the response result.

Aggregation

Dynamic aggregation from collected data is fundamental in some contexts, as it allows additional information to be extracted from the raw data. In our case study, being able to provide aggregation functionalities directly through the data retrieval process saves time and resources. At the same time, an optimized solution with reasonable timeframes is necessary to ensure its use.

The results obtained from the tests performed on the two different approaches are shown in Fig. 3.8. Although, as is obvious to expect, aggregation performed directly on the collected data performs comparably on both solutions, the use of pre-aggregated data in cases where the required aggregation granularity is greater than or equal to that of the bucket causes Advanced Bucketing to perform significantly better than the Time series collection approach, with the difference becoming more significant the further away from the data collection frequency the desired granularity is. This is because the use of the buckets pre-aggregation allows the result times to scale by an order directly proportional to the order of the bucket

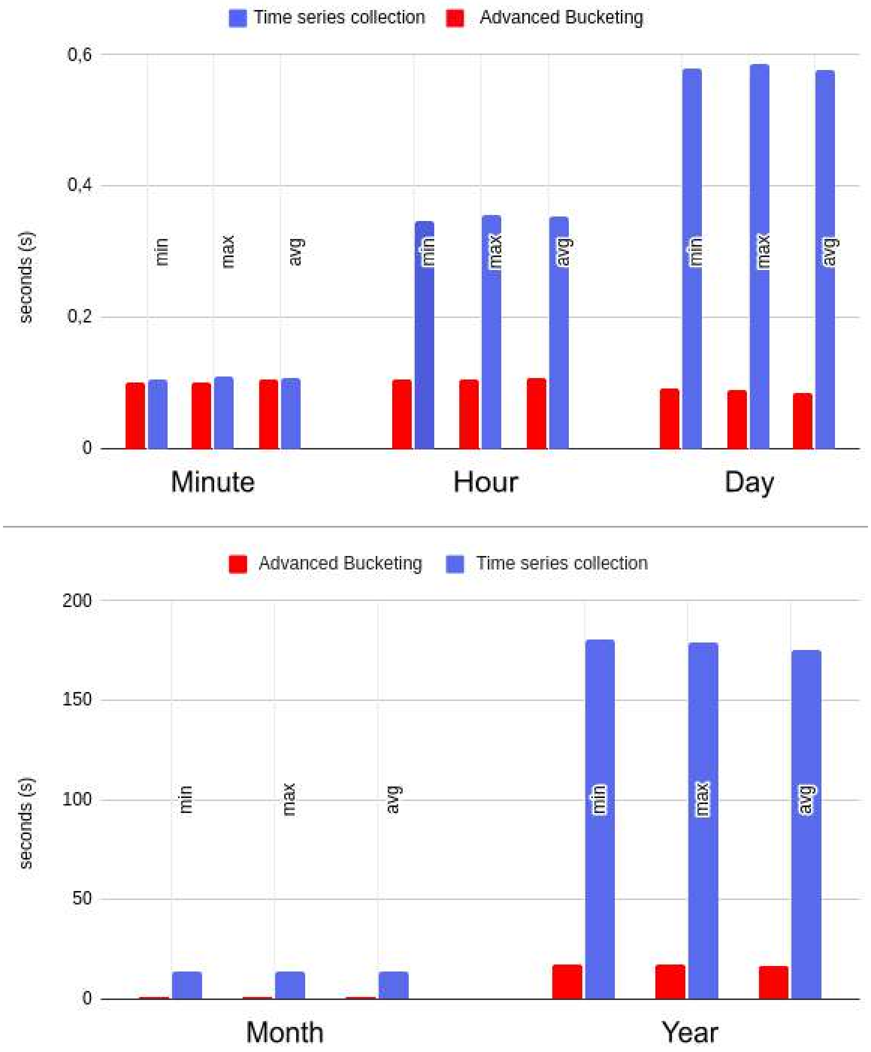


Figure 3.8: Comparison of data aggregation considering different granularities (minute, hour, day, month, year).

size. On the other hand, there appears to be no detectable difference between the different operations for calculating the minimum, maximum and average values.

Total count

Finally, the calculation of the total number of measurements present in a request becomes very useful in practical use, as it allows to know the amount of data required without having to first complete the retrieval of the data itself through paging operations.

In this case, as we can see in Fig. 3.9, the native MongoDB functions suffer considerably as soon as the number of data to be considered increases, making the use of the parallel structure built in Advanced Bucketing very useful. As in paging, in fact, the computation time required in the case of the Time series approach grows linearly as the number of data increases, while the behavior of Advanced Bucketing remains constant. Obviously the creation of the parallel structure necessary to calculate the value of *totalCount* depends on an initial processing time, which in this case is about 58s, and an additional insertion complexity as the structure itself needs to be updated.

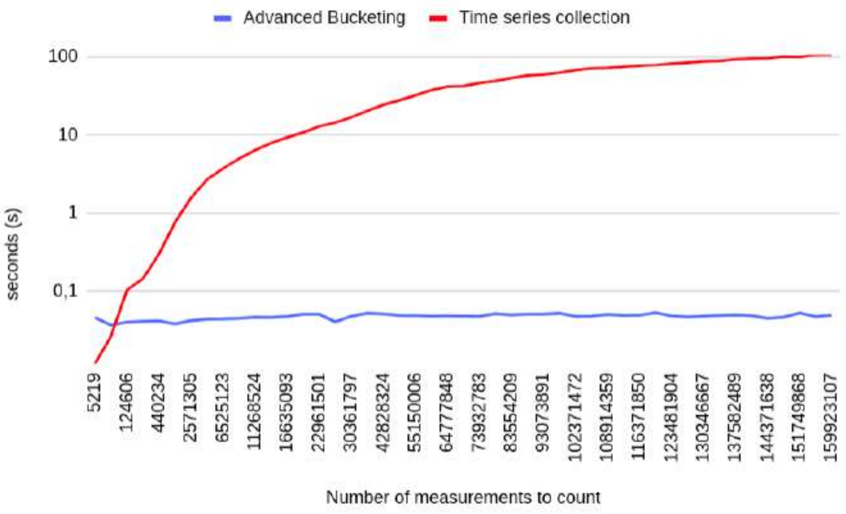


Figure 3.9: Comparison in logarithm scale of total measurements count varying the amount of data requested.

3.1.5 Final remark

This section presents two different methodologies that can be adopted for managing time series data on MongoDB. This approach should be noted with regard to smart city contexts in which, in addition to sharing large amounts of data, it is necessary to be able to quickly find and analyze measurements. The use of the bucket, together with other techniques, has been optimized to the maximum (keeping an eye on usability aspects), allowing to compare the results obtained with those provided by the standard methods applied to the new collections of time series. implemented in MongoDB 5.0.

Although the insertion times and the required development complexity are in favor of the data series management used in MongoDB 5.0, the Advanced Bucketing approach is preferable, considering also the current limitations of time series collections, in contexts where large amounts of historical data are required and therefore the analysis refers to wide time intervals. The results obtained from the operations of aggregation and management of the total count bring to the solution presented an additional added value which, when useful, can be decisive.

However, it is correct to underline that the approach used in the management of the

Time series collection approach was the basic one and it is not forbidden to apply various advanced methodologies to increase performance and optimize functionalities. In a future scenario it is therefore possible to think about deepening the use of time series collections and re-evaluate the two approaches in the different contexts of use.

3.2 Disruptive Technology for Smart Mobility in Real Use Case

Today's cities are facing a revolutionary era in urban mobility; this is due to different factors, among the others their continuous growth and the concentration of human activities. To prevent and solve problems related to mobility such as traffic congestion and air pollution (for instance due to $PM_{2.5}$) and its potential link with other risk factors (e.g. Covid-19 spread, as envisaged in recent studies [36], [37]), cities are in continuous search of adequate mobility solutions to satisfy the demand of the growing population, both living in or moving around the cities every day. As a result, decision-makers have to face more and more complex challenges when managing and planning mobility, combining new forms of mobility, that must coexist in the urban structure of modern cities, in compliance with the well-being of citizens and protection of the environment.

The concrete adoption of disruptive technologies in the decision-making processes can represent the pivoting point for a paradigm change in the management of mobility. Decision Support Systems, Artificial Intelligence, predictive algorithms, simulation models, Big Data analytics, etc. offer the opportunity to analyse the current mobility situation, identify present and future trends allowing to predict potential future mobility scenarios [38], [39]. Our investigation focuses on four European cities distributed in four different countries: Amsterdam, Bilbao, Helsinki, and Messina. Each of them offers a different perspective on urban mobility, in terms of characteristics, offered services and challenges.

3.2.1 European Use Cases

Amsterdam

Amsterdam, the capital of the Netherlands, in recent years has been growing rapidly in terms of inhabitants and visitors; this growth leads to increased mobility and traffic issues. The city has complex traffic streams with massive amounts of bicycles combined with cars and public transport; this drives the need for finding solutions that can conciliate the ever-growing use of bikes with the other means of transportation (from public transportation to private cars) resulting in more sustainable mobility for the whole city. Part of this view is a strategy tending to increase the appeal of bikes as the main mobility option [40]. This strategy goes through the improvement of the city network of bike lanes and of the overall cycling experience within the city, encouraging virtuous behaviours (e.g. respect of traffic lights) to avoid potential discomfort.

What Amsterdam is aiming for To reach these objectives the city of Amsterdam would like to align the mobility policies to the real needs of bike mobility, realise a data-driven decision mechanism, strengthen the safety and comfort of cycling, and encourage citizens to make sustainable mobility choices.

The role of disruptive technologies in Amsterdam From a broader perspective, a unique point to access data coming from different sources can support the decision-makers in the identification of the required information, reducing the time spent to search it and speeding up the decision-making process. Since different departments of the municipality (i.e. civil servants) are involved in decision-making, the possibility to easily share information among them (such as data, results of analysis/simulations, map layers, charts, graphs) would improve collaboration and overcome inefficient communication and silos, allowing at the same time the reduction of policy fragmentation and the subsequent uncertainties. From a more specific perspective, data analysis tools can support decision-makers in understanding different aspects of bike mobility (through the analysis of bike-related data) and in identifying dependencies among factors that could influence directly or indirectly bike mobility and its adoption. In this sense tools and models to simulate how decisions and policies can potentially impact on traffic and mobility would offer predictions and the possibility to compare different scenarios. This would allow decision-makers to make choices with minimal negative impact and to minimize related costs. Finally, effective visualisation of information is essential; a dashboard offering map layers, charts and graphs that summarise the status of bike mobility in the city would allow decision-makers to have, in a single view, the overall and relevant information they need to gain new insights about bike mobility in the city (e.g. type of road infrastructure/ bike paths, road safety level, traffic mix/sources, congested routes, cleaner routes in terms of air quality, greener routes, faster routes).

Bilbao

With an area of 41,60 km^2 and around 355.000 inhabitants, Bilbao is the heart of a metropolitan area that extends along the estuary of the Nervioon River with a population close to 1 million people. In the last 25 years, Bilbao has suffered an important urban transformation from an industrial economy to a city based on a service economy. This has helped to balance the city and provide a friendly environment for pedestrians with wider pavements, reduction of on-street car parking in the city centre, traffic light control system to cater for pedestrians and promenades for walking and cycling. Today, 65% of internal movements are produced on foot. In this context, the Sustainable Urban Mobility Plan (SUMP) [41] in Bilbao plays a significant role; its main objectives are:

- Reducing air and noise pollution.
- Improving safety by reducing accidents and fatalities.
- Guaranteeing universal accessibility.
- Improving energy and transport (passengers and goods) efficiency.
- Contributing to improve the attractiveness and environmental quality of the city.

Of particular interest is the "Pedestrian mobility strategy" aiming to promote non-motorized modes of transport (especially pedestrian displacement) since these best suit the sustainable mobility objectives. Part of this strategy is the transformation of Moyua plaza, for exclusive use of public transport, pedestrians, and cyclists, prohibiting private traffic.

What Bilbao is aiming for To reach its objectives, the city of Bilbao aims to obtain a global vision of the city in terms of sustainable mobility, to take decisions based on updated data (predicting the impact resulting from applied measures), follow a more agile decision-making process (facilitating communication between stakeholders involved in the definition and development of the SUMP), translate measures impact into health and life quality indicators and access data coming from scattered sources that is automatically collected and integrated.

The role of disruptive technologies in Bilbao In the context of Bilbao, it is essential that decision-makers can easily access the most updated data; in this sense tools that facilitate the connection of data sources and the data harmonisation (leveraging common and well-defined data models) would support decision-makers in their daily activities. Once data is collected, a data catalogue (as a unique point of access to the data) would offer the capabilities to search data considering different criteria; among them, the possibility to filter the available data by for example the “transport mode" would allow the decision-makers to reduce the time they spend to identify the data they need. Facilitated setup and execution of simulations (for instance, to forecast impact on traffic, mobility patterns or SUMP’s KPIs resulting from a measure/policy applied) would support the decision-making process reducing the time spent in performing those simulations. Tools to create charts and graphs that summarise the status of mobility in the city from the sustainability point of view would allow the decision-maker to have, in a single view, the overall and relevant information to globally monitor the mobility in the city. On the other hand, the possibility to define and create customised KPIs and indicators would allow the decision-makers to fine-tune the dashboards with all the relevant information that they need to take into account in the planning of the mobility in the city. To this aim, checking if the data is updated would allow the creation of analyses and simulations based on correct information that represents the real status of the city, whereas pre-processing of collected data would reduce the time needed to setup the analysis and simulation for decision-making processes.

Helsinki

Helsinki, the capital of Finland, is a continuously evolving and developing city. In this sense a particular example is represented by the Jatkasaari area. The shore area of Jatkasaari, literally meaning "Dockers Island", was previously used for industrial and harbour purposes; now it has gradually transformed itself into a residential area offering workplaces and services. At the same time, Jatkasaari is also a growing passenger and transport harbour due to its location (right adjacent to the centre of Helsinki). The harbour is the main connection between Helsinki and Tallinn, with growing mobility and opening of a new terminal in 2017. Annually 1 million private cars travel on the connection where a single main road leads in and out of Jatkasaari. This road feeds directly to the largest car commuting junction (70.000 cars daily) from the city centre to the western suburbs of Helsinki, creating interference. The Jatkasaari area is emblematic of the overall development Helsinki is facing, in particular, concerning mobility. In this context, to correctly cope with this evolution, the City of Helsinki’s traffic planning and traffic management need up-to-date and high-quality traffic information to support data-driven decision making. In addition, proactive and forward-looking approach is needed as the population of the metropolitan area grows and traffic situation changes.

What Helsinki is aiming for In this context, the City of Helsinki aims to check the status of traffic and its development, analyse how traffic could evolve, perform traffic forecasts, simulate traffic planning and land use, check the development and implementation of new infrastructures and policies, develop a master plan for city development (e.g. land use, mobility, housing). To reach these objectives it is essential to establish a unique view and understanding among traffic planning and urban planning, allowing the exchange of information among different departments (overcoming information silos). In doing so, the city of Helsinki faces some issues related to the availability of different map layers with different information representations moving from a department to another, the lack of people with competences for demanding analysis, the lack of time to get deep understanding of data and problems related to obtain raw data to be analysed with external tools.

The role of disruptive technologies in Helsinki A data catalogue as unique point of access that brings under the same umbrella the data produced by different departments would simplify the discovery and access of needed data, avoiding complications caused by scattered repositories managed by different departments of the same organisation. The data catalogue could leverage tools for the integration with existing ICT software and applications. This would allow on the one hand, the automatic check of information (e.g. automatic detection of inconsistencies in the data, such as missing mandatory fields, infringement of time constraints about updates) and on the other hand, the automation of repetitive tasks (e.g. extract relevant information and provide it in a more usable manner). Leveraging the data made accessible it would be possible to define pre-packaged simulations that need only minor operations to be executed (e.g. few parameters and/or initial input data). This would simplify the use of this kind of technology by personnel without specific competencies and skills who would be able to set up an entire simulation from scratch, and reduce the time needed and the acceptance of this technology, since the personnel will not spend too much time to learn how to use it.

Messina

The metropolitan area of Messina is one of the most extended urban areas in the south of Italy and the first in Sicily and counts over 620.000 citizens. In the city of Messina alone, there are over 250.000 inhabitants and most of them are commuters between Sicily and Calabria regions. Geographical peculiarities (the geographical shape of the city of Messina is stretched for 32 km beside the Tirrenian sea, and tight between its hills and the sea) and its role of main connection point between Sicily and the Italian peninsula have a huge impact on mobility in the city of Messina. The local transport system of the city consists of sea transport (hydrofoil and ferry boats fleets) and land transport (buses, tramway and rail transports network), operated by public and private companies. One of the main issues that affects both kinds of services (sea and land transport) is the lack of interoperability among the different departments of the Municipality that are involved for different reasons in the management of the mobility.

What Messina is aiming for Concerning mobility, the main challenge of the city of Messina for the upcoming years is twofold: on the one hand, to build mobility services able to fulfil the needs of citizens, dwellers, commuters and visitors, allowing them to move around and through the city seamlessly; on the other hand, the challenge consists in optimising the management and interaction among the different mobility and monitoring systems and services available in the urban area of the city of Messina reducing the waste of resources and costs for the Public Administration. A particular attention is paid on light mobility (e.g. extension of the cycle network with new bike-lanes and links between the centre and suburbs zones of the city to spread the use of bicycle mobility [42]) and pedestrians (definition of an integrated system of pedestrian areas and paths).

The role of disruptive technologies in Messina The different Departments of the Municipality would benefit of a unique data-access point to their data, avoiding the complication generated by the need of accessing scattered data sources (for instance, in the case of data hosted and managed in different repositories for the different departments). This would simplify the discovery of and access to the data needed by the decision-makers. In this context, tools to facilitate the connection to data sources (also from third parties) are vital. Data is the fuel of any activity related to analysis, simulation and the more information is available (not only in terms of amount but also in terms of variety), the more accurate and precise can these analysis and simulations be. In this context, advanced smart devices and virtual devices [43] (abstracted component characterized by specific high-level functionalities) offer the chance to access the needed information with the most appropriate frequency and accuracy, avoiding information overload and allowing a more efficient computation. In the management of urban mobility, analysis and simulations would support decision-makers in the identification of potential solutions (such as multimodal paths and possible intervention to increase public safety) [44] and hidden problems (such as related to public transportation and for planning maintenance interventions of road and public transportation vehicles).

Customisable dashboards to represent the information a decision-maker needs would allow to obtain a clearer view of the status of mobility, supporting the decision-making process in the most appropriate manner. Finally, the possibility to share information (such as data, results of analysis/simulations, map layers, charts, graphs) with people working in the same or a different department would improve the collaboration and the efficiency of the decision-making process, overcoming inefficient communication and information silos.

3.2.2 Final Remark

Despite their specific peculiarities such as organisational approaches and mobility needs to be satisfied, the cities of Amsterdam, Bilbao, Helsinki, and Messina have some commonalities in terms of potential application of disruptive technologies that can help their decision-making processes. The main aspect that emerged is related to the need of data, as a vital element to perform any decision-making activity; in this sense, it is important to underline that here the need is related to the easiness of accessing the data, that in most of the cases is scattered, or represented using different data structures with non-uniform standards. Uniform access to the data drives to another common point among the four cities, that is the exploitation of the possibilities offered by simulation tools, in particular, to forecast and predict the impact of decisions taken on traffic and mobility (such as the building of a new road, the creation of a LTZ). This kind of technology would allow the decision-makers to better design mobility solutions and policies, giving them the possibility to tackle complex problems and evaluate the implications of new policies. The third common point is data visualisation. Accessed data and results obtained from simulations and data analysis must be visualised in an easy-to-understand manner, this includes not only the data visualisation, but also the possibility of creating customisable dashboards in which the decision makers can arrange the information they need and represent it according to their preferences. From the result here summarised, it is possible to clearly identify a chain of needs with their corresponding solutions. The first link of the chain is the need of **accessing data**. Here tools facilitating the connection to data sources and the integration with existing IT systems can offer a valuable solution to overcome information silos and to build a unique data-access point to available data, allowing also the harmonisation of the data thanks also to common and well-defined data models and highlight the relevant information reducing the time to find it. The second link of the chain is the **analysis of the data** made accessible through the previous step and the execution of the simulation. Here it is important to highlight that beyond the possibility to perform analysis and simulation, the availability of tools that simplify and reduce the time needed to set them up plays a key role. In this sense, pre-packaged simulations ready to use, that guide the users in their setup, and tools, that allow the creation of customised KPIs and indicators, represent an advantage for the decision-makers. The third and final link of the chain is the **data visualisation**. Here, tools (e.g. Wizards) guiding the users in the creation of charts, graphs, map layers, etc. offer the opportunity to speed up the decision-making process by reducing the time of interpreting and understating the information. At the same time, the possibility to visualise different data in the same view through customisable dashboards offers the chance to obtain a bird's-eye view of the information that is relevant for each decision-maker, according to their specific needs. Considering the reported results, a final consideration can be made; even if cities could be characterised by a different IT maturity level, the most suitable way to effectively improve mobility decision-making processes is not a single technology, but a combination of disruptive technologies, that glued together unlock their respective potentialities and benefits.

CHAPTER 4

Cyber Security in Data Acquisition Systems

The growth of Smart Cities has resulted in the introduction of devices belonging to the Internet of Things (IoT) class into physical environments. These devices allow you to collect data and provide services to users. The security on the Internet networks where data transits, and the devices themselves distributed throughout the territory (for example, traffic detectors for smart mobility services) are at risk of cyber-attack by malicious users. This chapter proposes solutions to deal with these risk situations. The adoption of non-certified or non-validated Internet of Things devices can either cause errors in the provision of services or expose the system to cyber attacks that invalidate the processing of information, allow the insertion of fake data, or even cause the interruption of the service. A generic malicious user can replace the isolated IoT device that collects traffic data in a periphery with a tampered one or make unauthorized access to the system. A widely adopted solution is to trust IoT devices through a Certification Authority (CA), however, the certification process is quite delicate because it could also be exposed to attacks. Our goals aim to address these challenges by proposing a new brokerage-based approach to creating reliable IoT services. This approach considers technologies like Blockchain ed Edge Computing and security protocols. Another aspect of this issue regards the Edge systems. Typically, due to their limited storage capacity, pieces of data are continuously exchanged with Cloud systems which store them in distributed DataBase Management System (DBMS). This scenario, known as Cloud/Edge Continuum, is critical from a data security point. In this case, we propose a solution based on Transparent Data Encryption (TDE) for encrypting database files. In this chapter, we propose a solution to encrypt the data locally at the Edge and transfer them to a distributed database over the Cloud. This approach allows us to perform queries directly on encrypted data over the Cloud and to retrieve them on the Edge for decryption. The presented solutions will be supported by experimental tests and the results will be discussed.

4.1 Trusted Ecosystem for IoT Device

Nowadays, the massive usage of Edge computing and Internet of Things (IoT) devices can cause the rising of a large number of services affected by errors and exposure to various cyber-attacks. Even legislators feel that this risk is increasing in the particular historical moment we are going through. The European Commission is revising the European Network and Information Systems (NIS) Directive, which came into force in 2016, aiming to guide the Member States in applying cybersecurity and data protection strategies. Cybersecurity is one of the European Commission's priorities even in response to the Covid-19 crisis, many companies and governments have suffered from cyber-attacks during the lockdown.

For these reasons, the European Commission supports cybersecurity projects with funding programs such as Horizon 2020, Connecting Europe Facility, InvestEU, and the recovery found for investments by the member states in cybersecurity. The same needs have emerged in various countries of the world wherein governments invest in research and financing of companies that deal with cybersecurity. Many studies have focused on different technological solutions based on certified and validated IoT devices. Still, it was never considered the solution of relying on a trusted Edge computing third party issuing certificates close to the device to be certified.

In this work we propose a supervised, trusted environment in a heterogeneous ICT system to insert trusted IoT devices. Most IoT devices with microcontrollers and/or microprocessors are easily attackable. In most cases, they are positioned in places accessible to all and can suffer attacks compromising their correct functioning. This solution allows us to have a strong degree of security in issuing certificates through a trusted entity, a Broker identified in a security administrator of a trusted technician of the organization for which the device is certified. The Broker, i.e., a physical entity, has to perform a double function:

- he/she is an intermediary between the Certification Authority (CA) and the IoT device;
- he/she has to provide a reliable network connection;

By definition, the IoT device is considered an untrusted entity, and without a certificate issued

and stored on it, it cannot be used for data and information acquisition. Initially, the IoT device is considered a virgin asset, unable to perform any function. It can only work as a hot-spot using its credentials to gain access to its private network.
In our work we present a new method for the secure use of IoT devices in a smart environment. The device, already programmed for a specific task, will start to work only after being recognized by the organization to which it belongs. The Broker will allow to ensure device recognition by acting as an intermediary between the cloud and the IoT device. We made this choice to prevent the non-validated IoT device from being inserted into an acquisition system and sending false data.

4.1.1 State of the Art

The rapid growth of the number of IoT devices applied in different scenarios has led to the review and analysis of untrusted devices' validation and certification techniques. Traditional techniques for the certification of untrusted IoT nodes can expose the system to different attacks that affect its correct functioning.
For this reason, to optimize the certification techniques of IoT devices, the research activities have shifted to other technologies. Among these, the authors in [45] propose a network system based on self-certifying IDs that allow converting the IoT device's IP into a validated and recognized ID in a trusted domain. In [46] the authors propose methods to guarantee the authentication of IoT devices on the network. A mechanism based on Secure Vaults is proposed, i.e., a set of keys that changes over time based on the amount of information that IoT devices exchange with each other. Furthermore, as explained in [47], a particular framework based on security protocols such as FIDO is used to guarantee the authenticity of IoT devices within a sensor network.
To create secure IoT devices, the authors in [48] propose an architectural solution based on a General Purpose Unit (GPU) associated with System on Chip (SoC) radio systems for sensor control. The SoC operates as an identity module dealing with authentication and generation of X.509 certificates. The authors in [49] provide a solution for validating and authenticating IoT devices. A QR_Code based mutual authentication protocol is presented to offer multiple levels of security in data acquisition systems. A cloud utility called Trust as a Service (TaaS) is introduced to manage the services of IoT devices, based on subjective trust protocols for managing services [50]. To make IoT devices reliable, at the architectural level, to limit the resulting attacks and damage, the authors in [51] present an implementation of the Trusted Platform Module and Unique Device Identifier to ensure the authentication of

the IoT device. Furthermore, special protocols are used to encrypt the acquired data and information concerning the devices themselves. Certificate management systems based on technology are used to manage data acquired from certified IoT devices. In these systems, the certificates are checked and not the data acquired by IoT [52] devices.
To ensure the proper working of IoT devices, the authors in [53] carried out a scalability analysis. Large-scale distributed trust models, Holochain, are proposed that address security concerns by checking data integrity. To authenticate the services and data acquired by IoT devices, trust anchors and certificates of the X.509 type are introduced, applicable to various distributed systems. But encryption systems are not adopted for the acquired data [54]. The authors in [55] propose a secure model that uses device endpoints and gateways for the authentication and authorization functions for IoT devices in the Industry. The model presented is based on an Intel IoT architecture for gateways. For the integration of IoT systems with technology, the concept of a proxy is introduced in [56] to which an IoT device can download the acquired information and data. Each IoT device is equipped with an SDK proxy that holds the identity and a private key, keeping control over the transitions. Based on the preliminary analysis carried out on the technologies adopted for the certification of IoT devices, it is clear that none of the certification strategies gives the system a strong level of security.
In this section, we propose a certification system for trusted IoT devices, in a specific domain, through the CA of its organization. The main effort of this work lies in the presence of a trusted third party, the Broker, who, using an Edge mobile device, can start and carry out all the certification operations. This is possible only in the proximity of the IoT node to be certified. The presence of a certified physical entity (a person) that acts as an intermediary between the untrusted IoT device and the CA system allows solving problems related to attacks such as men in the middle and, in general, cyber-attacks the correct functioning of automated systems.

4.1.2 Motivation

One of the main advantages of using certified IoT systems is to have reliable distributed environments to provide efficient and guaranteed services. In this historical period, the digitalization of services has become one of the main problems in acquiring, monitoring, analyzing, and controlling in different scenarios.

Reference Scenario

Our reference scenario considers different actors involved in using the IoT system, and we show an example in Figure 5.1.

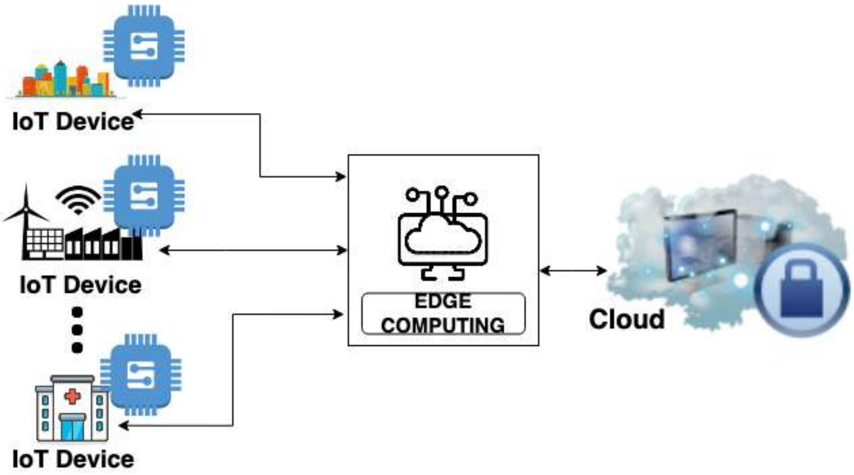


Figure 4.1: Reference Scenario.

The validation and certification process presented in this work is aimed at those IoT devices that are not disposable, easily compromised, located in areas accessible to all, and subject to numerous cyber attacks. The IoT device certification process takes place through Mobile Edge Computing (MEC). This certified mobile device can transport information between the single untrusted IoT device and the Cloud.
Only in the presence of a certificate issued by CA of the belonging organization, the device will send data, via a trusted internet connection, to a data centre in the Cloud that will process all the data and information collected. Certified IoT nodes can be used in various applications in Industry 4.0. For example, it is possible to provide a certificate and controlled access of specialized people within a refinery control room through this solution. Furthermore, these reliable IoT devices can be applied in data acquisition systems in the healthcare sector, where reliable data must be acquired through certified sensors. In special cases, this technique can be applied to certification automatic speech recognition devices to support dysarthria patients [57]. The IoT device certification process can be applied to municipal management systems to provide reliable digital services to citizens in a specific perspective of future Smart Cities.

Issues

In general, within a network of distributed IoT nodes, it is necessary to guarantee the integrity and confidentiality of the acquired data, as well as authentication for all the actors authorized to access and modify data and information [58]. Each IoT device must be uniquely identified as a single entity in the isolated domain and have a particular role within the accounting system. At this scope, it is important to define an authentication process to verify the entity's authenticity in the communication. The interaction of untrusted devices with backend systems can be subject to men in the middle attacks. These attacks can lead to false data and compromise the entire functioning of the sensor network. The system itself can be affected by attacks such as DDOS (Distributed Denial of Service), which can flood the backend system with requests to make it unreachable. The proposed solution uses an account management system, a PEP proxy, and public-key cryptography to block this type of attack by malicious users.

Requirement Analysis

To solve the problems that can afflict the system, it is necessary to design and implement specific algorithms and protocols to certify an IoT device in an untrusted environment. To certify a virgin IoT device, the presence of a third certified entity is required to act as an intermediary between the system CA and the device itself. As seen previously in the literature, there are different methods and procedures to certify a device IoT, but in the proposed system, a broker's presence is required. The Broker is a certified third party, which acts as an intermediary between the CA of the backend system and the untrusted IoT devices. The presence of an authorized and certified physical entity in the system is a solution that allows providing a strong level of security for the system. Automating processes and certifying IoT devices, through automated systems that do not require human presence, can lead the system to be exposed to attacks that can affect the system's correct functioning. The organization that manages the services and the entire distributed sensor network is equipped with its own CA. The CA must issue certificates both for authorized technicians and IoT devices inserted in the network. The authorized technician must perform as an intermediary, Broker, and provide a trusted connection for the IoT device's configuration processes. Furthermore, to avoid inserting a non-certified IoT device in the sensor network, we imagined proceeding with an ad-hoc configuration operation. Non-certified IoT devices can work as a hotspot and expose their credentials to access their private network to the

Broker.

Proposed Solution

This paper proposes an innovative solution to the problems related to the certification and validation processes of untrusted IoT devices. We presented a method for the certification of IoT devices that uses the presence, near the device itself, of a trusted third party. This entity is the Broker, a technician authorized and certified by the organization that manages the system and provides the services.
The Broker will be equipped with a MEC, a specially configured IoT device, or a smartphone and will play intermediary roles between the CA and the untrusted IoT device. The physical presence of the Broker allows providing strong security at the system. Although some tools allow certifying IoT devices, we want to propose a system that certifies the entire ecosystem in which the trusted IoT device is inserted. All the device certification processes will be supervised by authorized users (people) who have been entrusted with certifying and validating the devices. Every single isolated node of the distributed network, only in the presence of a CA certificate, will perform the task for which it was programmed and communicate securely with the backend.

4.1.3 Design

This section presents the method which seeks to contemplate the main issues highlighted above. The proposed method, shown in Figure 4.2, is composed of several components described below.

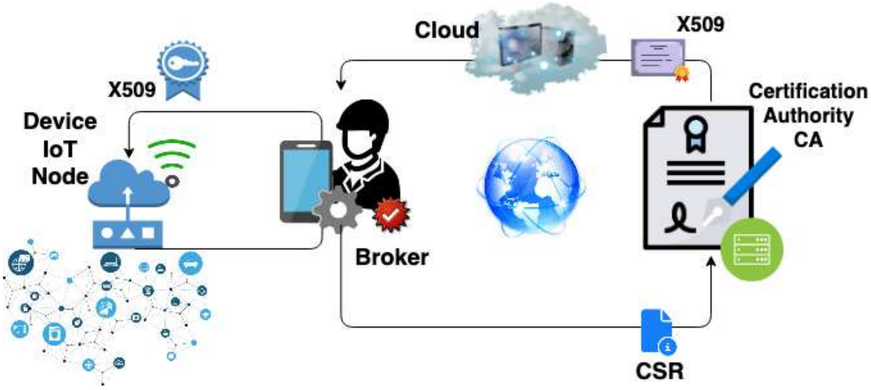


Figure 4.2: Main actors interaction.

The *Certification Authority (CA)* of an organization issues the certificates of trusted people, Brokers, and IoT devices to be inserted as nodes for data acquisition, control, and monitoring in different application scenarios.

The *Broker* is a trusted third party (person or group of people) and acts as an intermediary between the CA and the IoT device. The Broker must be equipped with a MEC. It can be a smartphone or a special IoT device, which works as an Edge mobile computing and performs all the certificate generation procedures to validate the untrusted IoT device. Besides, the Broker will initially provide a trusted internet connection to the IoT device to make appropriate initial configurations and test its correct functioning.

An *IoT Node* is an IoT device in a distributed network of trusted IoT device nodes. Only in the presence of a certificate issued by the CA of the organization to which it belongs will it be able to carry out the operations for which it was designed. The proposed system is based on a CA of the organization or company, which certifies and manages the entire distributed network of sensor nodes. This CA has to perform two particular functions: certify the Broker's identity, with the support of a system administrator, and certify the identity of each single IoT device in an isolated domain. To carry out any validation and certification operation of IoT devices, and to provide a trusted internet connection to them, each Broker must be authorized by the system. It is necessary to define a procedure for identifying and recognizing each organization's technician acting as a Broker.

We want to integrate into the proposed solution the use of Identity Access Management (IAM) [59], which manages system accesses using techniques based on role-based access control (RBAC). Each Broker has its account to which, during the registration phase, a particular role is associated that allows it to carry out certification operations for IoT devices.

To start the validation and certification procedure, the Broker must enter a password that only he knows and that is not stored in any volume storage in the system's backend. In the solution proposed, a physical mechanism is introduced to authorize the Broker functionality. To this end, the system must be equipped with an administrator responsible for authorizing its brokers. Only after the act in which the system administrator recognizes the Broker and authorizes his identity will start the offline process of generating the Broker certificate through the organization's CA. Once the CA issues the Broker's certificate, it will perform the functions for which it was identified. It is proposed that the untrusted IoT device initially performs the hotspot functionality. In general, we imagine the presence of a QR_Code with the access credentials to a private network exposed by the device once activated. Through this procedure, only the Broker near the device to be validated and certified can frame the QR_Code and connect to the device network to start the certification procedure.

This choice is made to solve the safety issues set out in the previous sections. With a valid certificate, the Broker must generate a pair of keys and send it over a private network to the IoT device. The IoT device received the Broker key pair, generates a CSR (Certificate signing request) containing its information, and will send it to the Broker. Once the Broker has obtained the certificate from the device and received a message with a positive result, it will be possible to change the network and connect to a secure internet connection to communicate with the organization's CA. The network switch from private connection to public connection is necessary to ensure non-certified devices within the system. Once the CA issues an X.509 certificate for the device, this is sent to the Broker, who will have to make a further network change to send the certificate itself to the IoT device. The Broker also attaches a configuration file containing all the internet connections deemed trusted for the system. Once the untrusted IoT device certification process has been completed, the device itself must be rebooted. To complete configuring the trusted and certified IoT device, the Broker must provide a secure internet connection to which the IoT device can connect. At this point, the IoT device will authenticate itself to the cloud system and be equipped with its accounting. The design of the architecture is based on microservices each of them implemented for performing specific tasks.

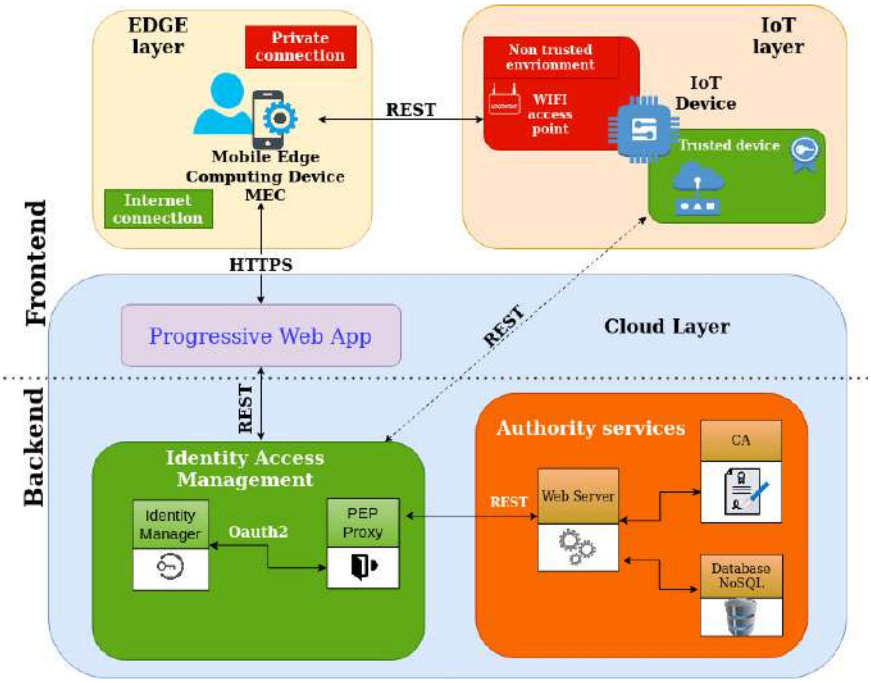


Figure 4.3: System architecture.

Concerning Figure 4.3, the deployed system manages both the certification processes of untrusted IoT devices and the real-time encrypted acquisitions from certified sensors. This choice allows the certification systems to be managed from IoT device nodes in such a way as to guarantee the integrity, authenticity, and truthfulness of the data acquired within distributed sensor networks.

The proposed architecture is characterized by three main macro-blocks: Edge Layer, IoT Layer, and Cloud Layer. The Edge Layer communicates with the Cloud Layer using a Progressive Web App, representing our system's front end. The Cloud Layer represents the system's backend and communicates with the IoT Layer and the Progressive Web App using the REST protocol. The Edge Layer allows users to interface with the system through a Mobile Edge Computing (MEC), represented by a Smartphone. The Edge Layer acts as a Broker between the untrusted IoT devices and the Cloud. Each IoT device exposes its private network to which the Broker can connect while communicating via a public internet network with the Cloud. The IoT Layer represents the generic IoT device in a specific domain. When started for the first time, being an untrusted device for the organization will work in hotspot mode, becoming a WIFI access point. When the Broker completes the validation and certification procedures, the IoT device will be recognized as trusted for the organization and will perform the functions for which it was created.

4.1.4 Implementation

In this section, we describe how the interactions of the main actors take place. In particular, we use appropriate sequence diagrams to formalize the resulting flows. Furthermore, the two main certification process are described.

Broker Certification Process

Figure 4.4 depicts the authentication and certification process for the Broker. The Broker must initially authenticate and log into the system's IAM (Figure 4.4 - Step 1). If the IAM recognizes the Broker, it will receive an Oauth2 token in response. Through the token, it is possible to trace information regarding the Broker's identity and role in the system itself. At this point, the Broker, to act as an intermediary for the IoT device certification process, must enter a password, which is not stored in the backend system. This password will allow the Broker to generate a pair of PGP keys on the browser necessary for its digital identity (Figure 4.4 - Step 3).

Through an offline process (Figure 4.4 - Step 4 - 5), the Broker must be recognized and physically identified by a system administrator. If the system administrator recognizes the Broker, he generates a CSR for him and sends the request to the CA (Figure 4.4 - Step 6). The

CA generates an X.509 certificate for the Broker, saves it, and associates it with the Broker's identity in the Cloud (Figure 4.4 - Step 8). Once the certificate has been obtained, the Broker is authorized to carry out the untrusted IoT devices' validation and certification operations.

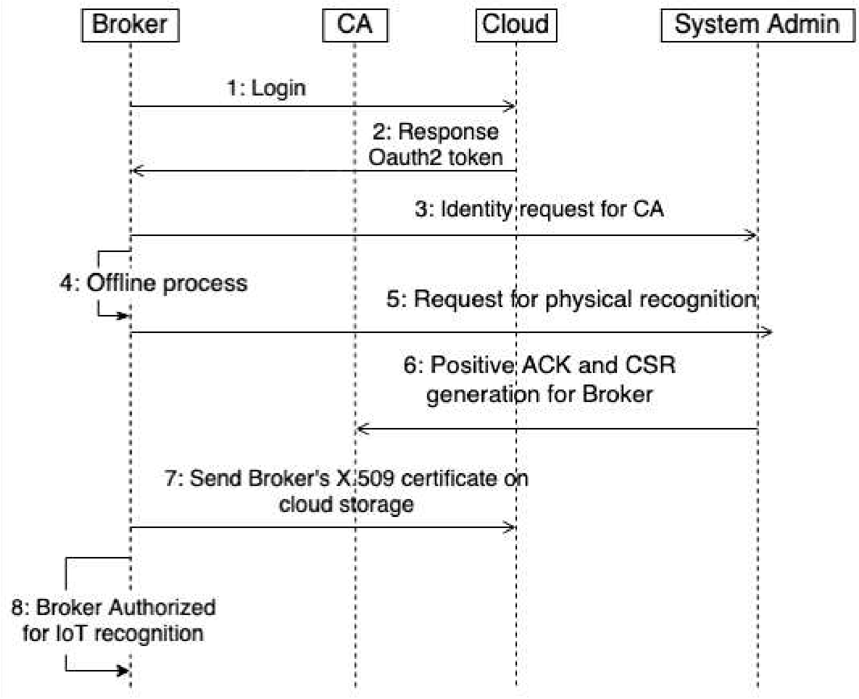


Figure 4.4: Broker certification process.

The communication between frontend and backend takes place via REST API authenticated by GE's Wilma. The Broker, after logging in, can enter the password that allows generating the PGP key pair. Once generated in the browser, the encrypted private key is sent to the backend. To activate the certificate request process, it is necessary to verify that there are no valid certificates for the Broker user who made the request. If the certificate is not present, a certificate request will be created even if an encrypted key was already present. If an encrypted key is present when a new request is made, it is replaced with the most recent one.

The second part of the process takes place offline. To ensure greater security, a network administrator must physically recognize the Broker who goes to his station. The administrator will see a list of certificate requests with the Brokers' data, from which the recognition will be done. If the Broker is recognized, then a CSR is generated. To customize and make the certificate that will then be issued "stronger", the CA signs a JSON containing the Broker's data and his Keyrock Id.

IoT Device Certification Process

Figure 4.5 depicts the certification process of an IoT device by a Broker. More precisely, the Broker must log in to the IAM of the system (Figure 4.5 - step 1) and obtain in response an Oauth2 type token. Once the token has been obtained, the CA verifies the Broker's certificate, and in the event of a positive response, it responds with a positive ACK message (Figure 4.5 - Steps 3 and 4). Besides the certificate verification phase, a check always takes place on a specific field of the certificate that contains a hash. The hash is the CA's signature on the technician's identity when issuing his certificate. It must always correspond to the one saved in the database and associated with the Broker. If the Broker's identity is certified, it can connect to the private network that the untrusted IoT device exposes (Figure 4.5 - Step 5) by scanning the QR_Code positioned on the device. The Broker will send a pair of RSA keys to the IoT device that it has specifically generated (Figure 4.5 - Step 6).
The device sends the CSR signed to the Broker with the pair of RSA keys received before by the Broker (Figure 4.5 - Step 7). To continue the IoT device certification process, the Broker must connect to a trusted internet connection, so it will have to change networks. Once connected to the internet, it is possible to send the CSR received from the IoT device to the CA (Figure 4.5 - Step 8). The CA will release the X.509 type certificate of the IoT device and send it to the Broker (Figure 4.5 - Step 9). The Broker must switch the network and connect to the private network that the device exposes (Figure 4.5 - Step 10). At this point, the Broker will send the X.509 certificate with a configuration file with the internet connections considered trusted for the system (Figure 4.5 - Step 11).
Once the certificate is obtained, the device reboots and connects to a trusted internet network (Figure 4.5 - Step 12). The Broker confirms the trusted IoT device's authentication process, storing the device certificate on the Cloud. Once the device certification process is completed, the IoT device logs in to the IAM system; if its identity is recognized, the IAM will respond with an Oauth2 token, (Figure 4.5 - Steps 13 - 14). Once the trusted IoT device's identity is recognized, and the token is received, the IoT device can perform the functions it was designed (Figure 4.5 - Step 16).

A Broker with a valid certificate must enter their password to access the certification functions. Entering the password allows generating an encrypted PGP key pair on the browser. At this point, the CA signature with Broker data is extracted from the DB, and its validity is checked. Only if the two signatures are the same, the system returns the Broker's encrypted private key to the frontend. If the browser can decrypt the encrypted private key

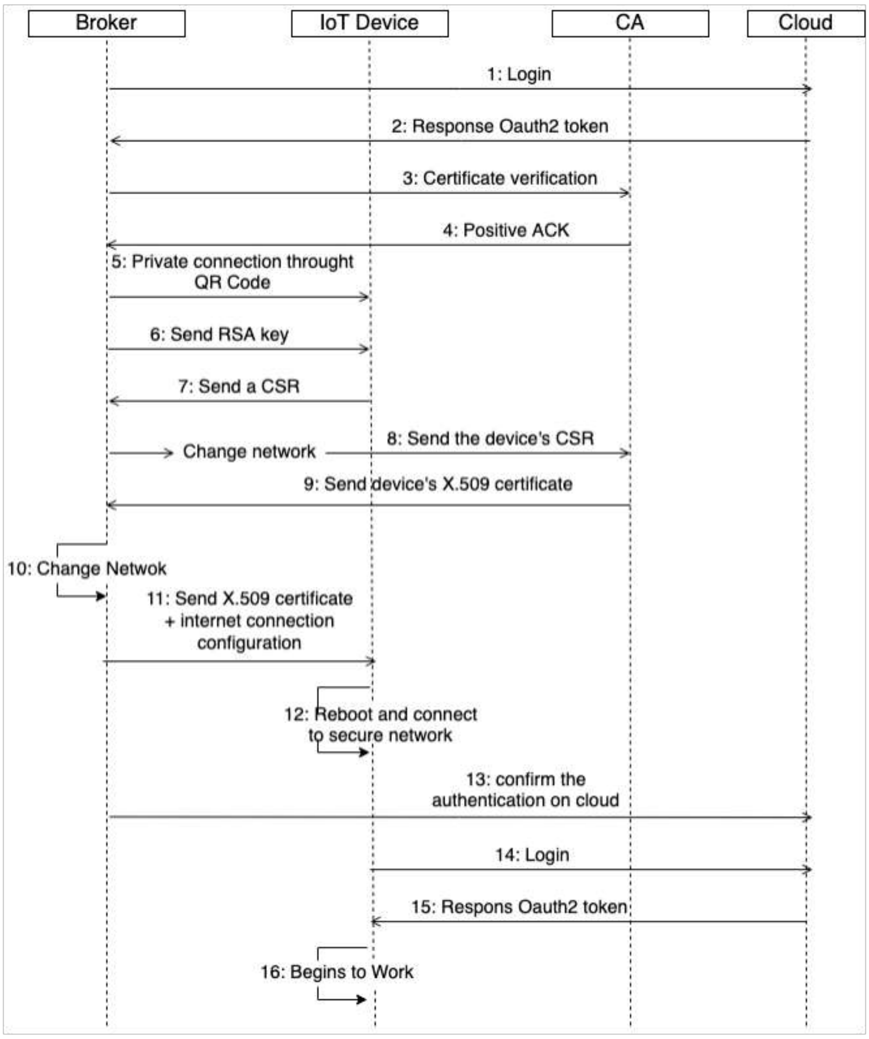

Broker
IoT Device
CA
Cloud
1: Login
2: Response Oauth2 token
3: Certificate verification
4: Positive ACK
5: Private connection throught QR Code
6: Send RSA key
7: Send a CSR
Change network
8: Send the device's CSR
9: Send device's X.509 certificate
10: Change Netwok
11: Send X.509 certificate + internet connection configuration
12: Reboot and connect to secure network
13: confirm the authentication on cloud
14: Login
15: Respons Oauth2 token
16: Begins to Work

Figure 4.5: IoT device certification process.

received with the Broker password, the device certification procedure starts. Through this procedure, it will be possible to allow the connection between the Broker's Mobile Edge Computing device (MEC) and the IoT device. The MEC generates an RSA key pair for the IoT device to which it is connected. The private RSA key pair is sent over the private connection to the IoT device. The IoT Device sends to the Broker its public key, its mac, and the CSR it has generated. The Broker connects to the internet and sends the CSR to the Cloud. This request is sent to the CA, which generates the X.509 certificate, returned to the Broker and the IoT device's connection configurations. The Broker who has successfully received the certificate must reconnect to the IoT device.

Meanwhile, the Broker creates and signs a JSON object containing the mac and its public key. It also signs the JSON containing the connection credentials and sends the X.509 certificate, the signed JSON, and the configurations for the IoT device's secure connection. On the one hand, the Broker, who must be close to the device, will provide an internet connection through his MEC. On the other hand, it will send the Cloud a request to confirm the IoT device's authorization process containing all the files saved on the device that will be stored on the DB. Having a connection, the device can configure itself with the backend and perform the functions for which it was created. We have developed a Progressive Web App for the simulation of the certification process of untrusted IoT devices. Regarding the procedures described in point A, the user can access a password entry function after logging into the IAM. By clicking on a register button, triggers the whole process described in point A, which takes place offline. For device validation, after logging in, the user will have access to an identity verification screen. With this function, the Broker enters the password with which it generated the PGP key pair required for the certificate request. If the checks on the password and its certificate are successful, the progressive App will launch a QR_Code reader feature. At this point, the Broker will be able to scan the QR_Code placed on the device and receive instructions to connect to it. When the Broker confirms the device validation with a click on the Web App, the process described in point B will start.

4.1.5 Validation

The evaluation's primary purpose is to quantify the system's performance and verify if it can be suitable for real-time use. Moreover, we also compared the algorithm's performance for the generation of key pairs in the browser, the algorithms for the generation of CSRs, and X.509 certificates for system users. To have truthful results, we performed 30 subsequent iterations for each experiment. All the graphs shown represent the average values of the 30 samples. To test the system's ability to work under stress, we use JMeter software. JMeter allows us to send a large number of requests simultaneously to the web-server. The maximum number of requests is fixed at 500. This number comes from the limit of maximum requests allowed by the proxy to protect the system from a single Ip address in a limited time interval. Furthermore, during the IoT device certification process (Fig. 4.5), we tested some requests made to the web server endpoints.

In particular, the flows considered critical for large-scale applications were tested: "3: The Broker Certificate validation", "8: The broker sends the Device's CSR", and "13: Confirm authentication on the Cloud". The choice to test these flows is made as they are automatic processes and not affected by problems due to the network connection or external factors.

In our experiments, we implement the features described in the previous sections using virtual machines hosted on the Garr network with nodes in Palermo (Italy) and a Raspberry Pi 4. The system's backend is located on a virtual machine with 4 GB RAM, 40 GB HD, and 2 VCPU Intel Xeon 64 bits, 2 GHz. In our system, the device IoT is a Raspberry Pi 4 with 4 GB RAM, a Broadcom BCM2711 processor, quad-core 64-bit ARM Cortex-A72 at 1.5 GHz. For the PGP key pairs generation, we evaluated the execution times for generating key pairs with passwords in the browser, for the Broker, using the OpenPGP protocol. We made a comparison between two algorithms for generating the PGP key pair: ECC (Elliptic Curve Cryptography) and RSA (Rivest–Shamir–Adleman). The simulations for generating passwords were carried out using sequences of 8, 10, 12, 15 characters.

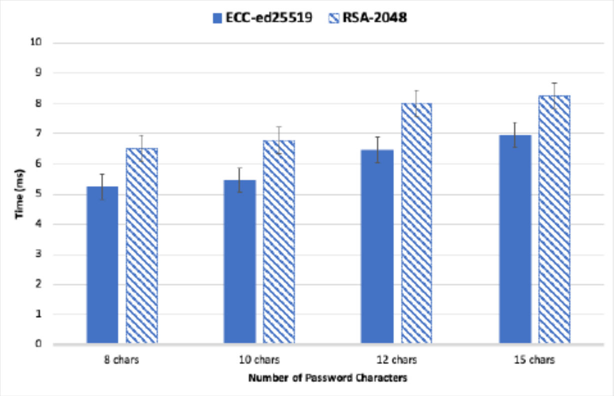


Figure 4.6: PGP Keypair Generation in Client.

Figure 4.6 depicts the average times for generating the key pair with a password in the browser. We observe that using an 8-character password, there is an average execution time of 5.24 ms with ECC, while with RSA, the result is equal to 6.51 ms. Using the ECC algorithm and a 10-character password, a pair of keys is generated in an average time of 5.45ms. While using the RSA algorithm, the key pair is generated in an average time of 6.79 ms. Instead, by entering a password of 12 characters, an average time of 6.46 ms for ECC and 8.01 ms for RSA is respectively used. By testing the entry of a password of 15 characters in length, we have an average time of 6.96 ms for ECC and 8.24 ms for RSA. To test algorithms' efficiency for the generation of CSRs and the issuance of X.509 certificates, we measure the average times for both the Broker (in the Cloud) and the IoT device.

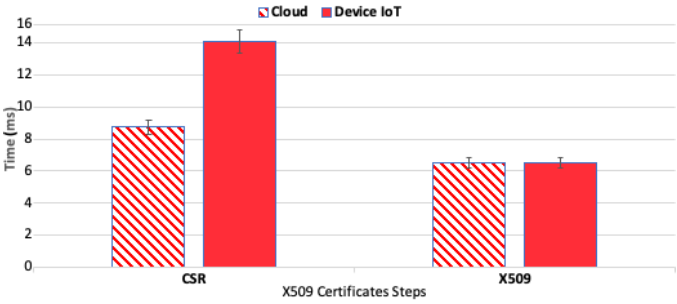


Figure 4.7: Certificates Generation.

Figure 4.7 depicts the comparison of the average execution times of the algorithms implemented for the generation of the X.509 certificate for the actors involved. We observe that the average time for generating the CSR in the Cloud is equal to 8.7 ms, while for the IoT device, it is equal to 14 ms. The generation of the X.509 certificate for the two actors takes place in an average time of 6.5 ms. In Figure 6.12 we report the increase in memory request on the machine in the interval corresponding to the three tested request flows.

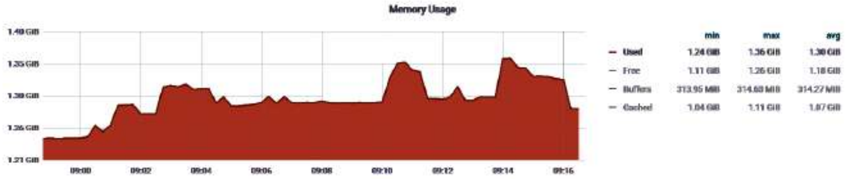


Figure 4.8: Memory usage request in case of flow of 500 get requests.

The use of memory is not affected by any problems as expected. No changes were reported on the use of the processor. For a more in-depth analysis, we monitored the use of resources by the container in which the PEP Proxy Wilma is running.

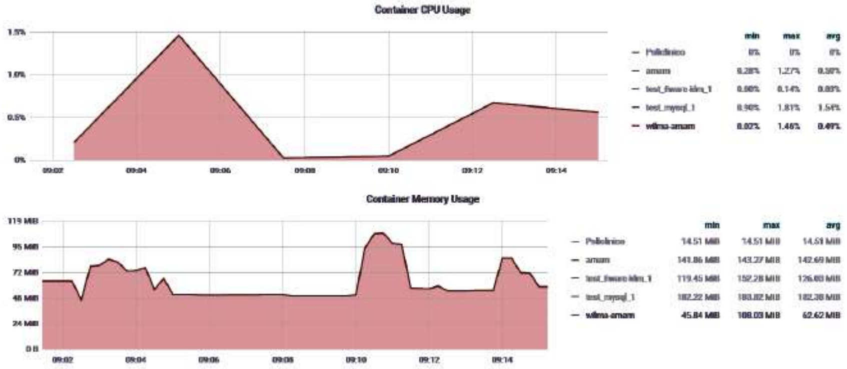


Figure 4.9: Resource monitoring for the Wilma Pep Proxy container in case of flow of 500 get requests.

In Figure 6.13 we report the use of memory and processor by the container. There are no relevant variations in the use of resources. The minimal and not significant variations in the use of resources allow us to say that the system, as expected, has a good performance under stress. There are no graphs on the use of database resources or other processes because there are no significant changes.

4.1.6 Final Remark

We investigated the current state of the art for IoT devices' certification in untrusted environments. Driven by rapid digitization processes, in different scenarios, we face new challenges related to the security of single nodes in distributed sensor networks through

IoT devices' massive use. Therefore, this work aimed to offer a system capable of limiting the damage due to cybersecurity attacks on IoT devices, accessible to all and positioned in unsafe places. In particular, the problems concerning different attacks have been solved, including brute force, men in the middle, and fishing or password theft, for Brokers. Another goal was to ensure the safety of users using distributed networks of trusted IoT devices. To understand the usability of the solution proposed, we tested the performance of the system. The development of the protocols and systems designed and the experimental analyses carried out helped us verify the system's feasibility. The experimental tests carried out emerged that the system can quickly generate pairs of keys with secure passwords. According to the X.509 standard, these processes allow generating certificates in a time that guarantees a good user experience to the system.

The stress tests did not reveal any significant critical issues. In general, the experiments made for different measurement depths highlighted outstanding performance in time, even compared to the most recent state of the art. To conclude, future work related to this research will focus on integrating this system into a real use case, using cryptographic protocols to increase the levels of security in data exchange between the various actors.

4.2 Certification of IoT Nodes in a Trusted Environment

In the last years, Internet of Things (IoT) devices are becoming very popular in many application fields, such as industrial installations, smart cities, smart healthcare services, domotic systems, and so on. According to recent statistics, the global IoT market will grow to 24.1 billion devices by 2030 [60]. IoT devices are small components with limited functionalities, low costs and often low or missing security features. Therefore, hackers could use IoT devices to conduct attacks to digital systems based on IoT devices. Several studies demonstrated that the 80% of connected devices are not able to ensure the privacy of personal information; the 80% do not require complex passwords, 70% use plain communication channels to send data, and 60% had multiple vulnerabilities in the user interfaces and firmware [61], [62]. To mitigate these threats several solutions based on honeypots or network traffic analysis have been proposed in literature [63, 64, 65]. The proposed solutions often employ machine and deep-learning algorithms to monitor the traffic, therefore they cannot be implemented in an environment with constrained computational capabilities.

Our work starts from the consideration that it is possible to certify the identity of IoT [66] by using a trusted Identity and Access Management (IAM) system that must recognize the devices and grant authorizations that allow them to access a service. In particular, we aim to certify IoT devices before they start sending data and taking part in the digital ecosystem they belong to. Then, all the IoT data are signed by the device to guarantee authenticity and integrity. However, the certification process can be a heavy task for IoT devices considering their limited computational and energy resources, due to the establishment of secure and remote communications with a Certification Authority (CA). Also, direct communication between an untrusted IoT node and the CA could suffer from a man-in-the-middle attack. In this paper, we propose a certification process for IoT devices based on Edge and distributed technologies, which support IoT certification building a trusted environment for the management of certificates. In particular, in our solution, we propose to enable an Edge gateway that allows technicians to certify IoT devices and services to verify their identity during their installation on the ground. We designed and implemented an innovative framework where the technician installs and digitally certifies the IoT device. The technician's digital equipment (e.g., a tablet or a laptop) is the Edge node in the certification process which decouples the communications between IoT and CA. After the Certificate Authority (CA) generates the IoT certificates, a Trusted Gateway stores signed certificates on InterPlanetary File System (IPFS) at the Edge to implement the IAM and check the identity of active IoT devices.

The main contributions of this scientific work are as follows.

- Design of an innovative framework for IoT device certification.
- Implementation of the proposed framework.
- Assessment of the performances of the developed system.

In our experiments, we analysed the response time of the system for certificating an Edge device and verifying its identity.

4.2.1 State of the Art

Different techniques have been proposed in the literature for the validation and certification of untrusted IoT devices. The problem encountered, however, is the exposure of Edge/Cloud systems to various cyber attacks that can compromise their functioning. In [67], the authors present a framework for validating IoT devices for large-scale automatic deployment. This secure certification methodology is based on security risk assessment and testing. The presented technique is applied to a specific scenario characterized by the Constrained Application Protocol (CoAP). However, the general significance of the study is not clear. This paper instead describes a generic solution applicable in different contexts such as urban mobility or the detection of environmental parameters. In [68], the certification process includes: risk analysis for a specific domain, identification of potential threats and vulnerabilities, security testing strategies, and execution of test suites. However, this technique requires time for the certification of the devices and is not very scalable in large scale. The work proposed in this paper aims to improve the aspect of scalability while improving security through the introduction of certification systems at different levels. There are also ways to certify insecure Edge/IoT devices using systems based on [45] self-certifying IDs. This mechanism converts the Edge device's Internet Protocol (IP) address into a valid ID in a domain that is trusted and recognized by the organization using it. This work is interesting but the connection to the network still introduces high levels of risk. The proposed approach instead provides that the Edge device connects to the network only in the presence of a valid certificate. In [50] the authors examine an Edge device management model based on the Trusted as a Service (TaaS) paradigm. This mechanism is able to manage all Edge device services that use subjective trust protocols. The Edge device certification process can be accomplished through the use of X.509 certificates [54], which are applied in distributed sensor environments, but there are no cryptographic functions for encrypting the captured data. Our system introduces the use of certificates but not only at the device level. In fact, it is important that the operator who installs or starts up the device is also certified. An approach based on operator certification is addressed in [2]. The authors address the issue of untrusted devices by proposing a certification protocol that uses a broker. This approach is tested to prove its effectiveness. In addition to the concept of brokering, the proposed paper introduces the use of the Blockchain in the certification process using IPFS (InterPlanetary File System) [69] for data storage. Blockchain can also play an important role in Edge device certification processes. In [17] the authors introduce the use of the Blockchain in systems that use certification systems based on brokering. However, the authors do not use the Blockchain for the certification process. The proposed work instead introduces a concept in which the Blockchain is primary in the certification process. However, Blockchain plays a vital role in the security process for Edge computing oriented applications. In many cases this technology is used for sharing certificates of trusted Edge/IoT devices [70, 71]. The advantages of the Blockchain in information security are described in [72]. The authors pay attention to safe privacy levels for smart ecosystem scenarios. In fact, it is evident that ensuring data integrity is essential to ensure the correct functioning of Edge/IoT devices in trusted distributed models in real-world scenarios. This aspect is addressed in [53]. Starting from these concepts, this paper proposes a system for the security of IoT devices that are certified using Blockchain technology. The system is tested demonstrating the effectiveness of the proposed approach.

4.2.2 System Architecure

The scenario we refer to in this paper considers a specific environment (e.g., an industrial plant, a district of a smart city,...) where several IoT devices are deployed to catch local data. Data are then sent to remote Cloud datacenters to be stored and processed. In this scenario, several attacks can be carried out [73]. For example, in the man-in-the-middle attack, the attacker is located in the middle of the communication between the IoT node and the Cloud and can send corrupted data to the Cloud. Spoofing is when an attacker impersonates a valid IoT device by sending fake data to the Cloud. Denial-of-Service (DoS) attacks can be performed to overload the remote storage and processing units in the cloud with a large amount of fake data or Distributed Denial-of-Service (DDoS) attacks can compromise the network connectivity. So, it is necessary to establish a trusted IoT-to-Cloud environment.

Even if several approaches can be used, certification represents the baseline to build a structured approach to mitigate security threats in IoT systems [74]. Figure 4.10 shows the architecture we propose in this paper to certify IoT devices and create a trusted distributed environment. It is characterized by different layers designed for specific functions: Edge Layer, Certification Layer, and Trusted Layer.

- *Edge Layer*: it is the architectural layer characterized by all the IoT devices installed inside an organization for different use cases. They can be microprocessors and/or microcontrollers for data collection and monitoring activities, and access remote services (i.e., storage and processing) using Edge nodes.
- *Certification Layer*: it is the certification layer characterized by trusted entities for issuing certificates for IoT devices in the Edge Layer; it includes key components, which are:
  - *Trusted Technician*: he is a staff member of the organization owner of IoT devices/-Cloud services, who is in charge for the certification of IoT devices, together with their installation and maintenance. The technician has a certificate issued by the CA.
  - *Certificate Authority*: it is the trusted node within the organization that issues certificates for IoT devices and signs the Certificate Signing Request (CSR) distributed by authorized technicians.

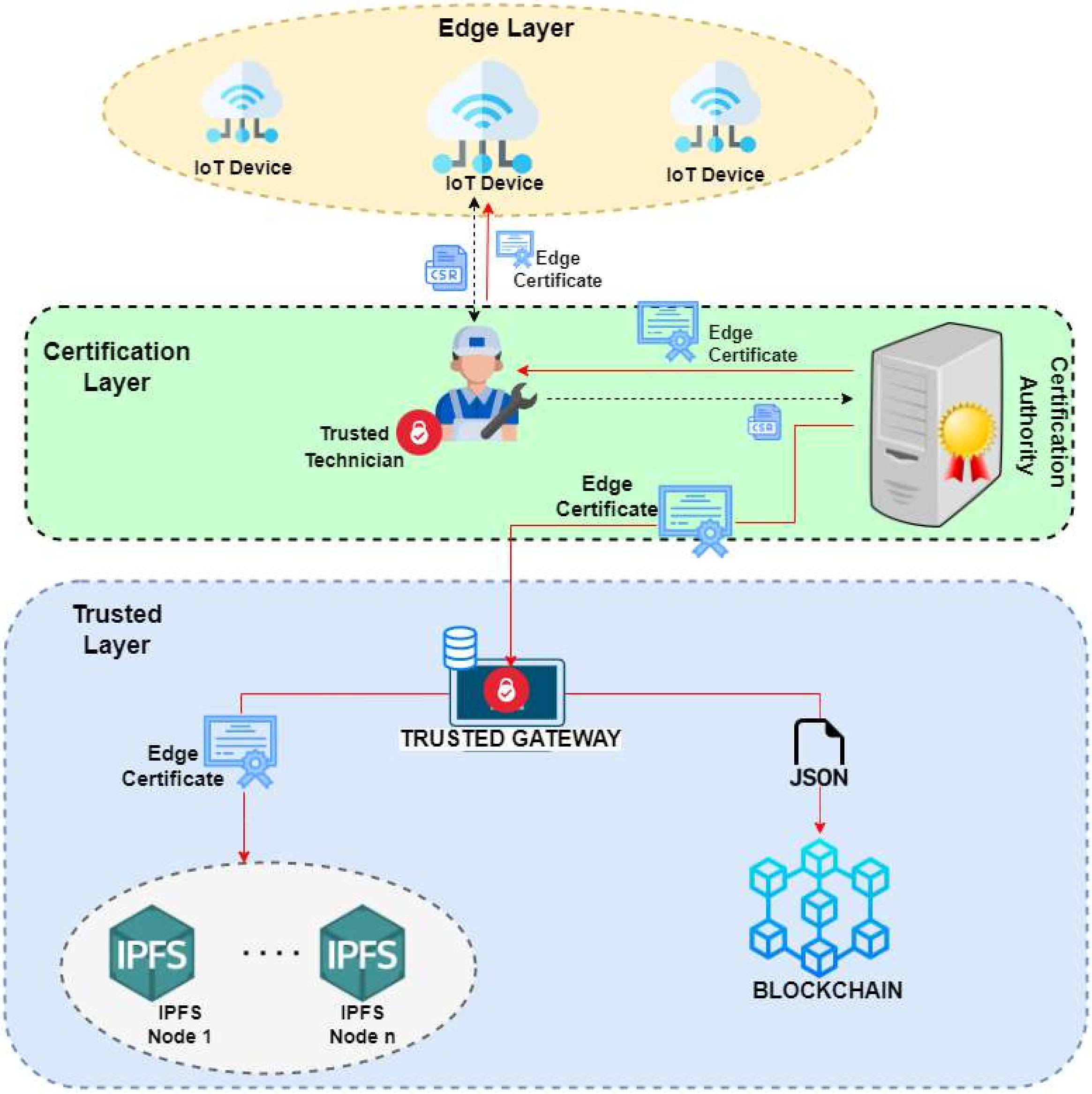


Figure 4.10: Reference Architecture.

- *Trusted Layer*: it is the architectural layer designed for the distribution and verification of certificates of IoT device at the Edge. It includes the Trusted Gateway, that is a certified Edge device designed to distribute the IoT Certificates into the IPFS private network and register the Certificate issue in the Blockchain.

The certification process of an IoT device is started by the Trusted Technician, whom computing device acts as a gateway between the IoT device and the CA. He goes close to the IoT device and set up a point-to-point communication with it. At the same time, the technician sets up the communication with the CA through a Virtual Private Network (VPN). More specifically, the Technician logs into the IoT device and generates a key pair, then he creates a CSR and finally, sends the CSR to the CA. The CA, checks the CSR integrity, generates and signs the IoT Certificate. The IoT Certificate is sent to the Trusted Gateway, which stores the Edge Certificates inside the IPFS distributed File System. To track the status of the trusted environment during the time with references of the nodes belonging to it, we have integrated the Blockchain into our system. In the Blockchain, for each released certificate, the Certification Identifier (CID) that is used to locate the certificates stored on the IPFS is registered. In particular, the Trusted Gateway register the CID whenever a certificate is released.

4.2.3 Implementation

In this section, we describe in detail the implementation of the certification process for IoT devices, with particular attention to two fundamental phases: 1) *the generation of the CSR* and 2) *the publication of the IoT certificate on IPFS and the Blockchain*.

CSR generation

For the certification of the IoT device, it is necessary that a technician authorized and certified by the organization starts the process for generating the CSR. To connect to the IoT device, the technician must use a point-to-point communication using available connections exposed by the IoT Device (e.g., Bluetooth, WiFi, USB...) (Figure 4.11 - Step 1). In the meanwhile, a local service is started in the IoT device for generating a key pair (Figure 4.11 - Steps 2 -3). Once connected, the technician starts the generation of the CSR for the IoT using information on the organization the IoT device belongs to, the IoT MAC address and its Public Key (Figure 4.11 - Step 4). To sign the CSR, the technician must generate a pair of RSA keys which he sends to the IoT device to which it is connected Figure 4.11 - Steps 5 - 6 - 7). Once all the information has been collected, the IoT generates and signs the CSR and sends it to the technician (Figure 4.11 - Steps: 8 - 9). At this point the technician, having obtained the CSR, switches the communication to the WAN port for a secure (VPN-based) remote communication with the CA and sends the CSR of the IoT Device to the CA (Figure 4.11 - Step 10). The CA verifies the signature of the CSR and, if the verification is successful, it generates the X.509 type certificate for the IoT device. The IoT Certificate is then sent to the Trusted Gateway and an acknowledge message is sent to the technician to inform him of the successful certification of the IoT device.

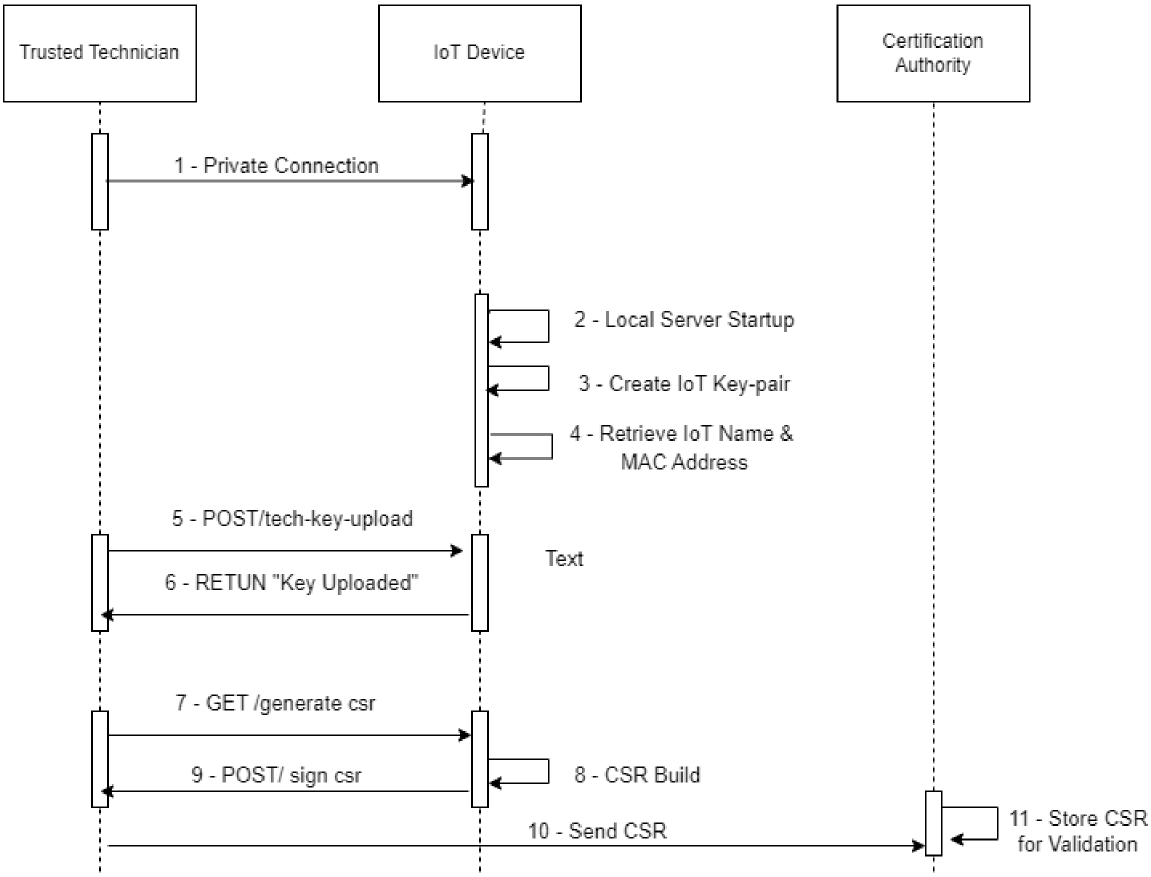


Figure 4.11: CSR generation process.

Management of IoT Certificates on IPFS and Blockchain

To ensure the integrity and non-repudiation of IoT device certificates, and to implement an IAM system at the Edge, we implemented a certificates storage within an IPFS network and their registration in the Blockchain.

Once the IoT Certificate has been issued by the CA, it is sent to the Trusted Gateway (Figure 4.12 - Step 1). The IoT Certificate is distributed in the IPFS file system, appropriately configured, so as to be accessible only from trusted nodes in the organization, such as Edge nodes that implement a check on the validity of IoT data (Figure 4.12 - Steps 2 - 3 - 4). Once the IoT certificate is upload in IPFS, the Trusted Gateway gets the CID, that is the hash code of the stored file used to retrieve it by IPFS, (Figure 4.12 - Steps 5 - 6). The Trusted Gateway proceeds saving the CID on the Blockchain (Figure 4.12 - Steps 7 -8). To implement this process, we used a testing instance of Ethereum [75] and simulated the generation of a Smart Contract containing the information on both the Trusted Gateway and the CID. In this way, it is possible to trace all the IoT devices certified within the Organization.

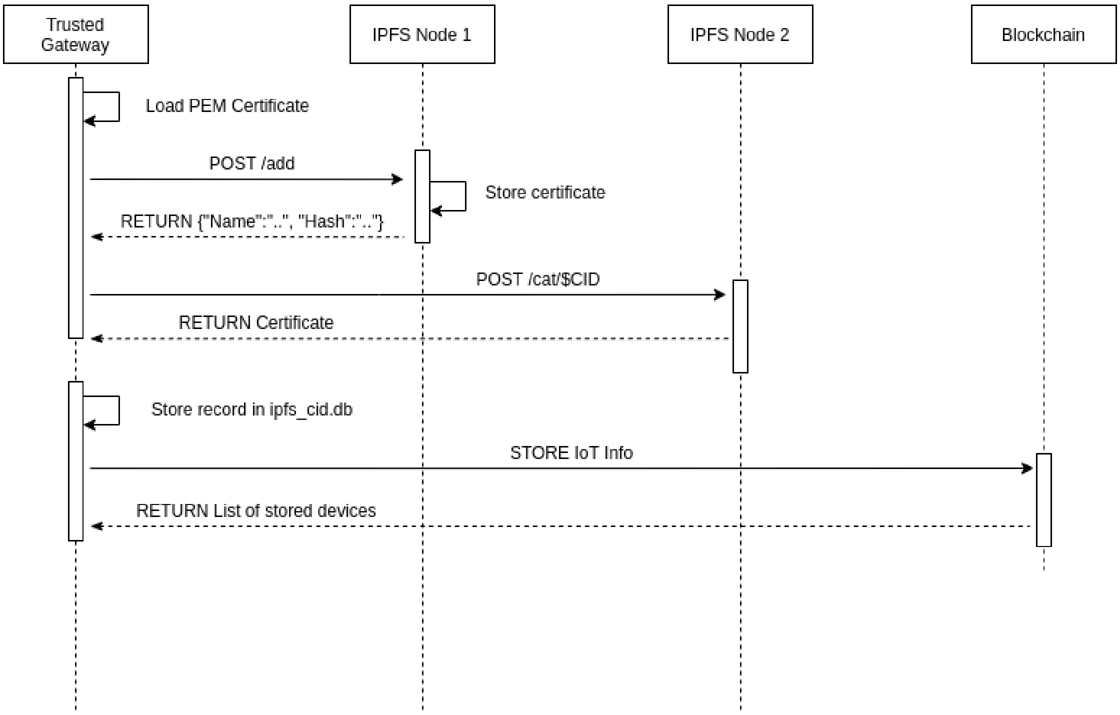


Figure 4.12: Management of IoT certificates.

4.2.4 Experimental Results

The validation of the system took place through a series of tests. The purpose of the tests is to demonstrate the system's ability to implement the described flows. Each experiment was repeated 50 times. The tests relating to the generation of the CSR and the storage and recovery process of the IoT certificates from the IPFS are reported. Furthermore, the process on the Blockchain registration was simulated. The IoT devices has been implemented with a Raspberry Pi 2 Model b with 1GB of RAM and a 900MHz ARM Cortex-A7 processor. The Trusted Gateway has been implemented with a Raspberry Pi 3 Model B+ with 1GB of RAM and a processor ARMv8 Cortex-A53 64bit at 1.4GHz. The first test refers to the process of CSR generation.

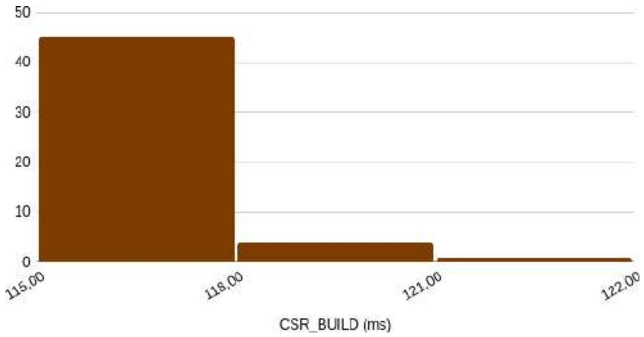


Figure 4.13: CSR generation time at the Edge.

Figure 4.13 shows that the process of generating the CSR with the IoT device data and the signature by the authorized technician takes an average time of 117 ms.

The second test concerns the storage of IoT Certificates on the IPFS network. The certificate obtained from the CA is temporarily stored in the memory of the Trusted Gateway which starts the process of uploading them to the IPFS nodes.

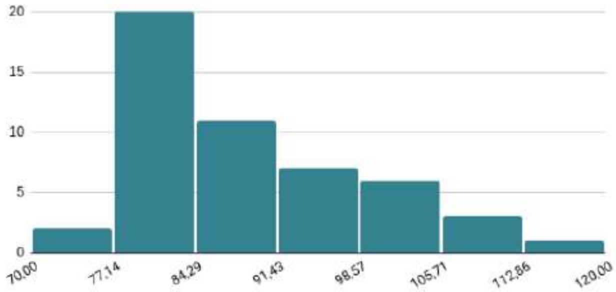


Figure 4.14: Average IoT Certificate publishing time on IPFS.

Figure 4.14 shows the average storage time of the IoT certificate evaluated on every single phase of the process represented in Figure 4.14. The average time for the upload of IoT certificates on IPFS is 90 ms. The third test verifies the feasibility of our system with reference to the average times to retrieve an IoT certificate from the IPFS network.

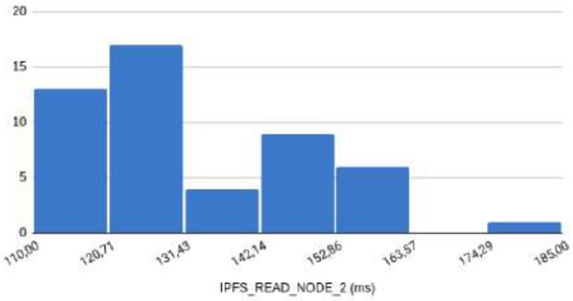


Figure 4.15: Average IPFS retrieval time.

Figure 4.15 shows the average certificate retrieval time from the IPFS network equal to 133ms. As can be seen, recovery times are not always homogeneous, precisely because they depend on the behavior of the network and the propagation times of the certificate. Fig. 4.16 shows an experiment to test the ability of the trusted gateway to verify flows and interact with the Blockchain for IoT certificate CID extraction. The flow concerns Figure 4.12 - Steps 7 - 8. As the reader can observe, the average time required to complete this process is 635.55 ms.

4.2.5 Final Remarks

We address a delicate issue concerning the certification processes of IoT devices present in different reference scenarios. Particular attention is paid to the processes for generating

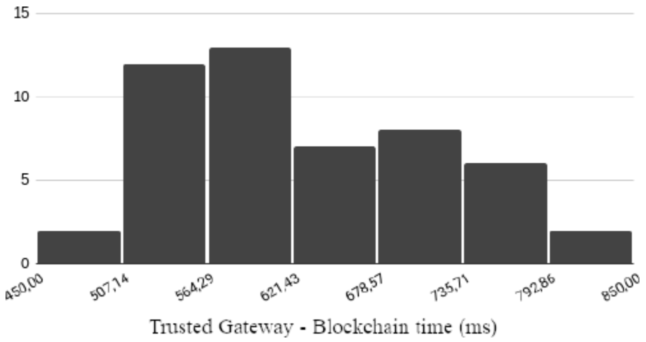


Figure 4.16: Trusted Gateway Performance

the IoT certificate issued by a CA and to provide an additional level of security within the system with the Trusted gateway, which is in charge of saving the certificates within the IPFS system and storing the CID in the Blockchain. In this way, it will be possible to check the validity of IoT data by checking the identity of the IoT device and verifying its signature on data using certificates saved on IPFS. If the signature is valid then the device is considered a trusted entity, otherwise, it is blacklisted. Performance experiments allowed us to prove the robustness of the proposed solution. However, from a future perspective, it is interesting to evaluate the resistance to the various cyber-attacks to which the system could be exposed. In this context, it is also possible to evaluate which key generation algorithm is more suitable for the proposed scenario. As regards the role of the Blockchain, however, it is interesting to evaluate its role as CA in future work.

4.2.6 Data Security in Cloud/Edge Continuum

Cloud Computing has been an innovative and widely used paradigm due to its considerable advantages. It allows the end-user to efficiently manage Information and Communication Technology (ICT) resources in an easier way abstracting and providing them as Infrastructure, Platform, and Software services. The use of this approach was disruptive, but the abstraction provided to the end user has introduced some important weaknesses:

- security issues related to Cloud providers that can potentially access end users' data;
- bandwidth exploited for accessing the remote services; and
- overhead introduced by a centralized approach that can affect real-time services.

The Edge computing paradigm has emerged to partially mitigate the aforementioned Cloud issues. This innovative approach involves the use of tiny devices (e.g., microcontrollers and/or microprocessors) to process data closer to the place where it is generated and collected with the purpose to change the typical centralised Cloud approach. However, the Edge is not conceived to perform Big Data processing that, sometimes, has to be moved to the Cloud. Therefore, according to the specific system requirements, micro-services could be moved from the Edge to the Cloud and vice versa. This paradigm is referred as Cloud/Edge Continuum [76]. In this research work, we consider a Cloud/Edge continuum system scenario able to manage sensor logging Big Data in a secure fashion as shown in Figure 11.1. For this reason, a peer-to-peer, NoSQL document-oriented, and Transparent Data Encryption (TDE) enabled DataBase Management System (DBMS) is strongly required. Specifically, a DBMS for a Cloud/Edge Continuum environment must be: *peer-to-pee* because it has to be executed in both the Cloud and unreliable Edge devices; *NoSQL document-oriented* to simplify the management of sensor logging Big Data according to a schema-free approach; and *TDE-enabled* to store the file system in a secure fashion. Nevertheless, at the time of writing this paper, a DBMS providing all these features does not exist. For example, by analysing the major NoSQL DBMS solutions, Cassandra is peer-to-peer and provides TDE, but it is column-oriented instead of document-oriented; MongoDB is document-oriented and provides TDE, but it adopts a master-slave instead of a peer-to-peer architecture; etc.

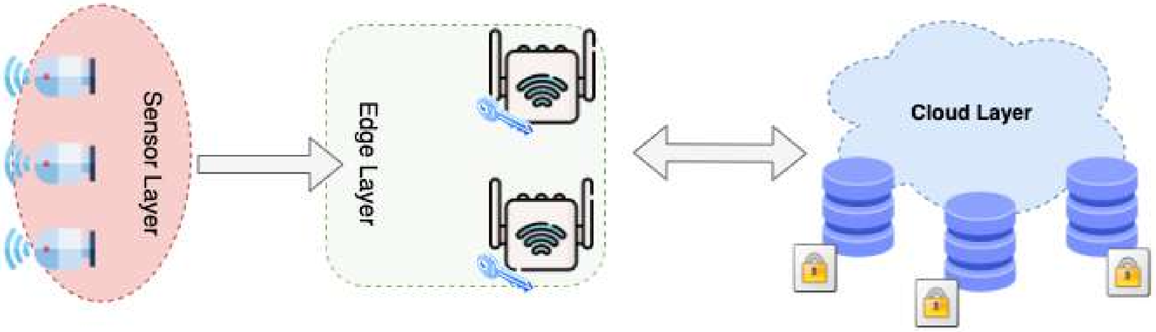


Figure 4.17: TDE Application: Reference Scenario.

To overcome all the aforementioned DBMS requirements, in this paper, we start with CouchDB, i.e., a NoSQL document-oriented DBMS adopting a peer-to-peer data distribution approach in which all the cluster nodes are synchronized even though they are not always up and ready. Specifically, it is easy to be deployed in a cluster in which it is not required the continuous connection of nodes that own a part of the whole data and that are synchronized when they are up and reachable. This approach is in contrast to other alternative NoSQL document-oriented databases such as MongoDB in which the distribution of data is implemented through the master-slave approach. Nevertheless, CouchDB does not support TDE. Therefore, this paper aims at overcoming such a gap.

4.2.7 Related Work

Nowadays, the number of Cloud/Edge continuum scenarios (e.g., in smart cities, healthcare,urban mobility, finance, etc.) is continuously increasing. Data security is one of the most crucial aspects to focus on when developing Cloud/Edge systems. In particular, considering Edge computing, data vulnerability is ever greater [77]. Various aspects concerning the security of Edge and/or Internet of Things (IoT) devices have been addressed in the literature [78], referring to complex certification processes [2] from various points of view. Sometimes, especially for large-scale deployments, expensive solutions in terms of technology and/or business are not feasible for companies [17]. It is necessary to implement actions that allow secure data collection and exchange in Cloud/Edge systems even in the case of large deployments how, for example, in a smart city where thousands of Edge devices are scattered around it and they exchange data with each other ones and with Cloud systems [79]. In this case, it is possible to think about encrypting the data stored in such systems. In [80], the authors focus on the positive aspects of using TDE on a standard database by assessing how good security performance degrades system performance. The TDE is often used to improve the security of relational DBMS(s) [81]. In [82] the authors remark that using this technique the database will hit a CPU and storage performance overhead. However, on Cloud systems, this situation can be managed. The authors demonstrate that the impact of managing security aspects can be optimized by considering CPU, I/O, and RAM performance. The purpose of our paper is to use this technology in a Cloud/Edge continuum scenario by evaluating its performance. With the spread of Big Data, security has become the major issue. In addition to concerning the acquisition systems [12], it is a problem of archiving software. In [83] an analysis is done to evaluate how the open-source Hadoop Distributed File System (HDFS) is used to store huge amounts of data with high throughput and fault tolerance. However, the security model was not designed and has become the main drawback of Hadoop. In terms of storage, metadata and data security is a problem for HDFS. The authors in their work explain the importance of this issue by explaining how companies are showing only a few layers of security in Hadoop such as Kerberos and TDE which is, therefore, a technology also used in the corporate world. These problems are now common also in Edge computing and, for this reason, we want to investigate the use of TDE in Cloud/Edge continuum environments.

In [84] the authors focus on how relational databases provide built-in security controls and mechanisms. The problem, which is most evident in Edge environments, is that information residing in the data store is at great risk. In their work, the authors try to introduce a new level of security. The used approach consists of segregating information based on its level of sensitivity and dynamically creating referential integrity constraints during the execution. For example, the primary keys of the restructured tables and the attributes of the most critical information were protected using the TDE utility provided by Oracle to prohibit the illegitimate use of information. The authors measured the querying performance of this approach. Our work aims at improving performance even while maintaining a solution of the same type. In [85] the authors note that the solutions for TDE provided by the major DBMS solutions are limited to protecting data at rest only and appear to be useless if the adversary has physical access to the server, which is a likely risk during the hosting at the Cloud or at the Edge. The authors propose an alternative approach to TDE based on an abstract model. This model takes into account the specific risks of the Cloud and extends the encryption to cover data in use and partial data in motion. Our research work, instead, aims at managing data in motion in a secure way.

The data security issue is addressed in [86], where the authors analyse and compare two current approaches: Oracle's TDE and the standard encryption provided by MySQL. Also interesting is the assessment that is made considering both single-server and distributed configurations. We intend to go beyond the approach of using TDE for MySQL databases and try to understand which technology is more functional in Edge/Cloud scenarios.

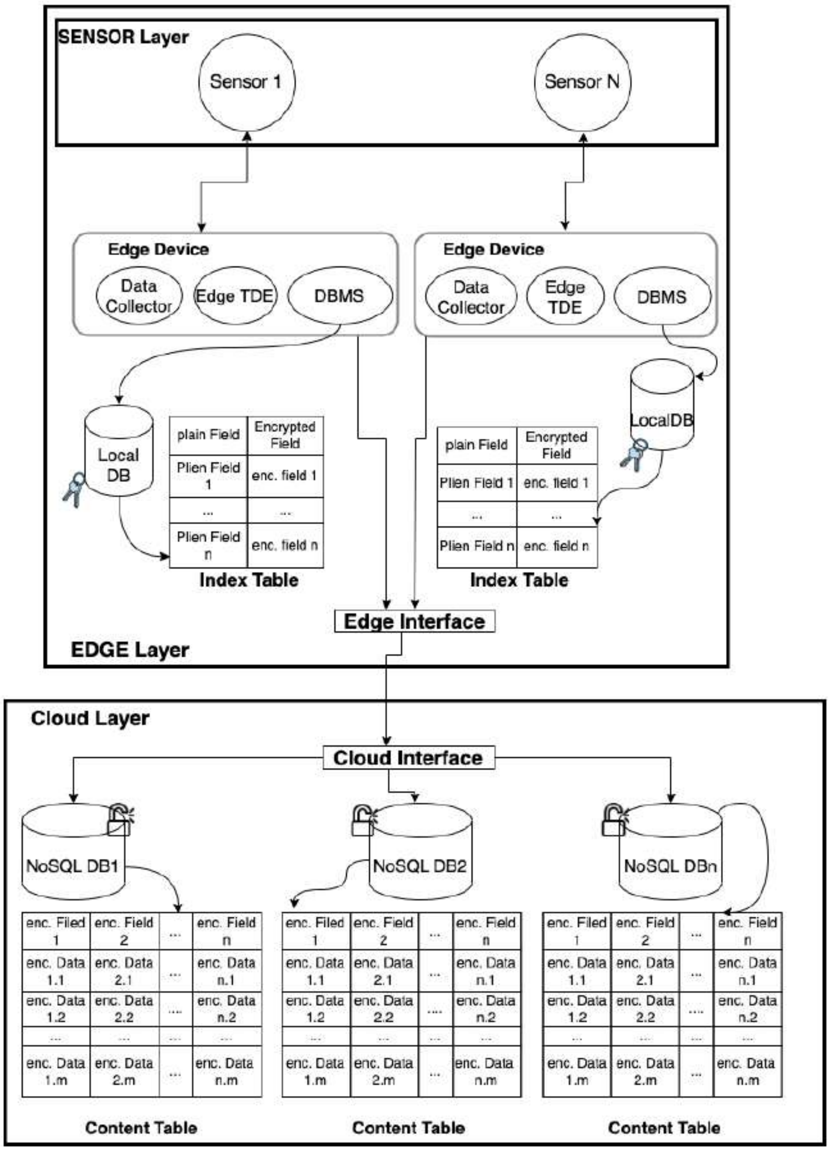
In fact, the work proposed in this section aims at introducing an innovative solution that uses the AES algorithms [87] to encrypt data without degrading system performance even in Edge environments [88, 89].

4.2.8 System Architecture

Figure 4.18 shows the system architecture designed for secure storage of sensor logging Big Data through TDE mechanisms in a Cloud/Edge continuum context. The architecture is divided into three macro-layers to allow a more detailed analysis of each element and its features.

The *Sensor layer* is characterized by sensors that detect and collect data from the environment in which they are installed. Every single sensor generates hundreds of logging data that are managed by the Edge layer.

The *Edge layer* includes microprocessor and/or microcontroller devices able to acquire and process the data generated by the sensors acting in the Sensor Layer. Inside the Edge layer, there are different software modules:



plain Field	Encrypted Field
Plien Field 1	enc. field 1
...	...
Plien Field n	enc. field n

Index Table

plain Field	Encrypted Field
Plien Field 1	enc. field 1
...	...
Plien Field n	enc. field n

Index Table

enc. Filed 1	enc. Field 2	...	enc. Field n
enc. Data 1.1	enc. Data 2.1	...	enc. Data n.1
enc. Data 1.2	enc. Data 2.2	....	enc. Data n.2
...	...	...	...
enc. Data 1.m	enc. Data 2.m	...	enc. Data n.m

Content Table

enc. Filed 1	enc. Field 2	...	enc. Field n
enc. Data 1.1	enc. Data 2.1	...	enc. Data n.1
enc. Data 1.2	enc. Data 2.2	....	enc. Data n.2
...	...	...	...
enc. Data 1.m	enc. Data 2.m	...	enc. Data n.m

Content Table

enc. Filed 1	enc. Field 2	...	enc. Field n
enc. Data 1.1	enc. Data 2.1	...	enc. Data n.1
enc. Data 1.2	enc. Data 2.2	....	enc. Data n.2
...	...	...	...
enc. Data 1.m	enc. Data 2.m	...	enc. Data n.m

Content Table

Figure 4.18: System Architecture.

- *Data Collector* is responsible to collect and process pieces of data coming from sensors.

- *Edge TDE* is responsible to start the TDE mechanism to encrypt the data acquired by the Data Collector;

- *DBMS* acts as an interface between the acquisition and storage system. The data acquired by the sensors are encrypted within the Edge Layer and transmitted to the Cloud system. Every single Edge device is equipped with a symmetric key used to encrypt the acquired data, and a Local database.

The *Cloud Layer* refers to all the software components running in the Cloud, which enable communication with the Edge devices of the Edge Layer. The Cloud only serves as a storage for encrypted data. In this way, any malicious user who attacks Cloud Storage systems can only recover the encrypted values and not obtain information on the adopted data model.

Particular attention has been paid to the design of TDE mechanisms for the secure archiving of logging sensor data in a Cloud/Edge Continuum context based on a NoSQL Database. The main idea is to memorize all the logging sensor data collected at the Edge into a Cloud storage system by adopting a TDE approach. As previously stated a NoSQL document-oriented, schema-free and peer-to-peer DBMS solution supporting TDE currently does not exist. For this reason, we focused on the design of TDE mechanism for a distributed database in a Cloud/Edge continuum environment. In the following, we describe how our TDE works. Every single sensor sends the plain data to the Edge device where the data encryption mechanism is started. The TDE uses the symmetric key encryption approach based on the AES protocol to encrypt data related to both field names and contents. Nevertheless, we had to face an issue paradoxically caused by the robustness of the AES protocol: if the same piece of data is encrypted twice with the same key, the result is different due to the presence of a random factor that allows increasing the security of data encryption. This means that if we encrypt a new field name in the Edge device and we store it in the distributed database deployed over the Cloud and if we subsequently try to compose and execute at run-time a query on the Edge device, we obtain an error because the two encrypted fields names corresponding to the same plain field (the one encrypted at the time of data storage over the Cloud and the one encrypted subsequently at the time of query composition and execution at the Edge) are different. This prevents the execution of queries. To solve this problem it was decided to create a simple mapping system for encrypted field names using an index table stored in the local database of the Edge device including the list of plain fields and their corresponding encrypted data.

In this way, the data model for the acquired logging sensor data can be dynamically built according to a NoSQL document-oriented and schema-free approach within the Edge Layer and the actual encrypted data can be transmitted to the distributed database deployed over the Cloud. In the Cloud database, the data (both field names and contents) is saved encrypted and no decryption mechanism can be foreseen. In this way, a malicious user can only acquire meaningless encrypted data in the absence of a decryption key. In case the end-user wants to obtain the data stored in the Cloud distributed database, it is necessary to compose queries using the encrypted field names stored in the Edge device's local database. This mechanism enables the development of a robust Cloud/Edge continuum ecosystem in terms of security that can prevent the classic attacks made on the storage systems.

4.2.9 Implementation

In this Section, we describe in detail the implementation of the TDE mechanism for data encryption on the Edge and its storage on a NoSQL DBMS deployed in a Cloud system. We have focused our work on two fundamental phases: 1) *Data acquisition, encryption and memorization* and 2) *Data retrieval and decryption*.

Data Acquisition, Encryption and Memorization

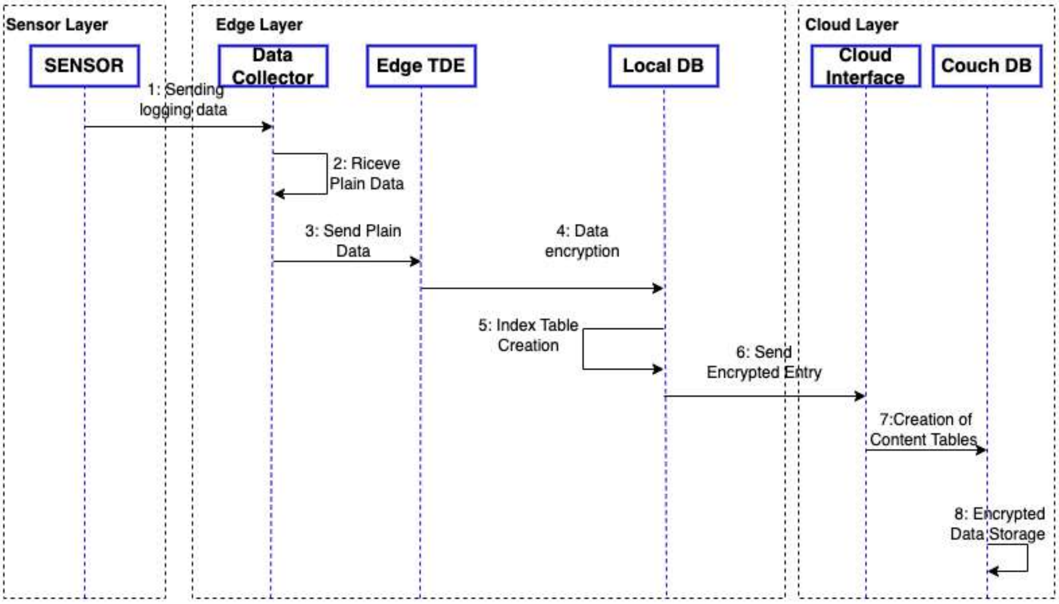


Figure 4.19: Data storage flow.

As a NoSQL database, the choice fell to CouchDB, a NoSQL document-oriented, schema-free and peer-to-peer solution. The main problem with using CouchDB is the lack of an integrated TDE mechanism: our objective is to fulfill such a gap. The logging pieces of data that are acquired by sensors are sent from the sensor layer to the Data Collector software module acting at the Edge Layer (Figure 4.19 - Steps: 1 - 2)). The Data Collector allows the Edge layer to acquire plain logging sensor data and send it the Edge TDE module (Figure 4.19 - Step 3), which implements the TDE mechanisms. Within the Edge TDE module, the plain logging sensor data is encrypted using the AES 256 symmetric encryption algorithm and sent to the Local DB (Figure 4.19 - Step 4)). To solve the problem due to the random factor of this algorithm, it was necessary to create an index table (Figure 4.19 - Step 5) that is used to map the correspondence between the plain fields names and the corresponding encrypted data. The index table is created in the local database present within each single Edge Device. The encrypted data is transmitted to the Cloud Interface module acting at the

Cloud Layer (Figure 4.19 - Step 6). This software module has been implemented to allow communications between the Cloud Layer and the Edge Layer, specifically allow the sending of encrypted data to Couch DB. To store the encrypted data on Couch DB it is necessary to create a Content Table containing both the encrypted fields names and also content data, i.e. the encrypted sensor logging data (Figure 4.19 - Step 7). By creating this table it is possible to save the encrypted sensor data on the Cloud (Figure 4.19 - Step 8).
This design choice was made to allow that sensor logging data can be saved in CouchDB in encrypted mode over the Cloud. As a consequence, in the event of an attack, the malicious user can access encrypted data. The mapping between plain fields name and corresponding encrypted data is contained within the Edge, therefore both encryption and decryption operations can be performed only at the Edge layer.

Data Retrieval and Decryption

For data retrieval and decryption, a query-based mechanism has been implemented at the Edge layer allowing secure data extraction from the Cloud. This design choice was made to allow an high level of security and reliability of the data. The authorized user can access the Edge device and retrieve the data stored in CouchDB over the Cloud. The user interacts with the Edge Layer to start the data retrieval and decryption process (Figure 4.20 - Step 1). The Edge layer of Couch DB requests the local Database to extract encrypted fields name

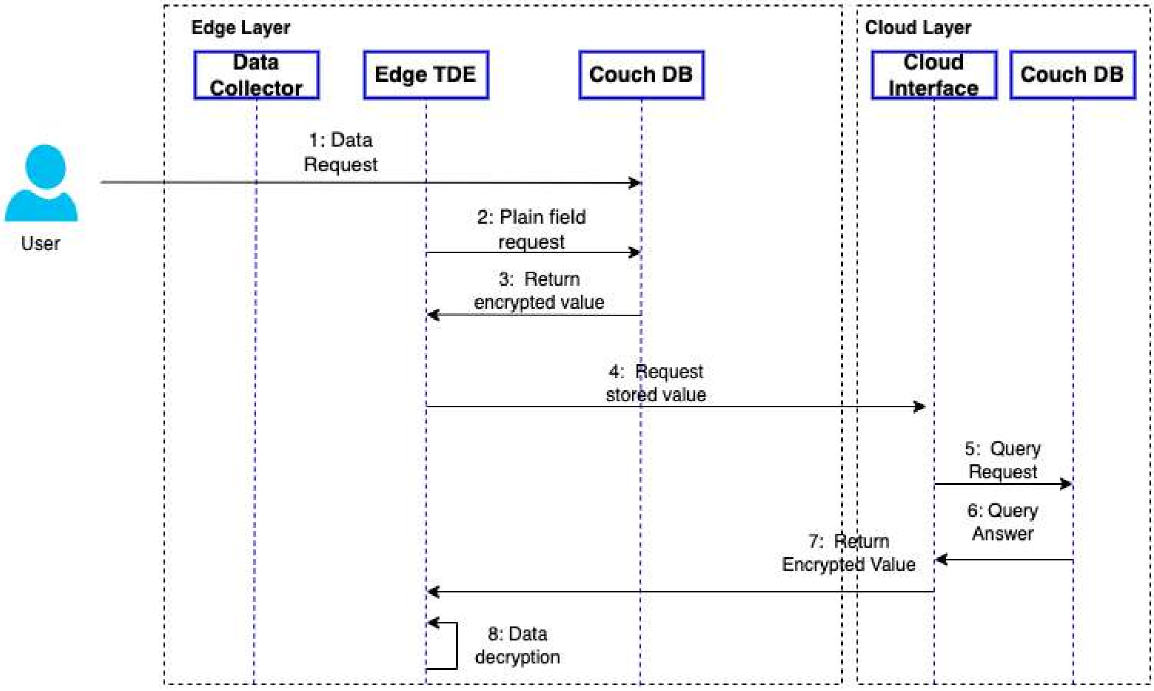


Figure 4.20: Data Decryption flow.

contained in the index table, (Figure 4.20 - Step 2). The encrypted fields names are returned to the Edge TDE software module where the data decryption processes take place. (Figure 4.20 - Step 3).
From the Edge TDE, a query is build considering encrypted fields names that is submitted to CouchDB acting at the Cloud Layer to extract the encrypted content data (Figure 4.20 - Step 4). In the Cloud layer, the Cloud Interface software module manages all the queries that are made from the Edge device to CouchDB. The Cloud Interface carries out the query on the CouchDB database which replies with the requested encrypted content data (Figure 4.20 - Steps: 5 - 6). At this point, having found the data requested by the user, the Cloud Interface sends the encrypted content data to the Edge TDE module running in the Edge device (Figure 4.20 - Step 7). The Edge TDE module contains the decryption symmetric key to decrypt the content data that are returned to the user (Figure 4.20 - Step 8).

4.2.10 Experimental Results

The experiments presented demonstrates the goodness of the described solution. In particular, a dataset containing logging sensor data, i.e., temperature, humidity, and ping times, was used. The dataset contains 130.338 records. Specifically, the dataset was divided into four parts of increasing size. A summary of the size breakdown of the considered dataset is reported in Table 4.1. The data has been encrypted and a dataset with encrypted field names has been created. For performance evaluation, projection queries were made. Three queries of increasing complexity were considered allowing us to compare the performance of the system using three of the major NoSQL DBMS, that are Cassandra, MongoDB, and CouchDB:

Dataset Dimension	Number of Records
25%	32.584
50%	65.169
75%	97.753
100%	130.338

Table 4.1: Experiments Dataset Partition

- first query: returns all documents that have the ping value equal to 17.28 seconds;
- second query: returns all documents that have the ping value equal to 17.28 seconds and a recorded temperature value greater than or equal to 22.00 degrees Celsius;

- third query: returns all documents which have ping value equal to 17.28 seconds, recorded temperature value greater than - equal to 22.00 degrees Celsius, and humidity value greater than 35%;

For the sake of clarity, we underline that the traditional relational DBMS (RDBMS) terminology changes with the considered NoSQL DBMS solutions: the concept of record or tuple typical of an RDBMS turns into a row (considering Cassandra) or document (considering MongoDB and CouchDB). The concept of a table changes into a collection considering MongoDB and CouchDB and so on. Furthermore, the performance of a mechanism that allows regenerating the database containing the encrypted field names has been evaluated. The tests were carried out on a Virtual Machine (VM) with the following hardware characteristics:

- Microprocessor: AMD® Ryzen 7 3700U, 4 cores;
- RAM: 8 GB;
- Hard drive type and capacity: NVMe SSD 128GB

Linux Ubuntu 20.04.03 LTS was installed on the VM along with:

- MongoDB: enterprise edition, version 5.0.6;
- DataStax Enterprise (Cassandra): Version 6.8.20;
- CouchDB: Version 3.2.1.

The software modules have been developed using Python 3.8.10.

Querying Time Performance

The querying response times allow us to understand if the proposed system can be used in real Cloud/Edge Continuum scenarios. Three tests were performed with increasing query complexity. In Figure 4.21 the results of the first query are reported, it is evident that MongoDB is the most efficient in terms of querying response times. In Figure 4.22 the results of the second query are reported. Also, in this case, MongoDB is the most efficient solution in terms of querying times. In Figure 4.24 the results of the third query are reported. Even in this case, MongoDB is the most efficient solution in terms of querying times. As highlighted, the results show MongoDB as the most performing solution. It is equally evident that for the Edge context, the other solutions are also usable. However, the only solution

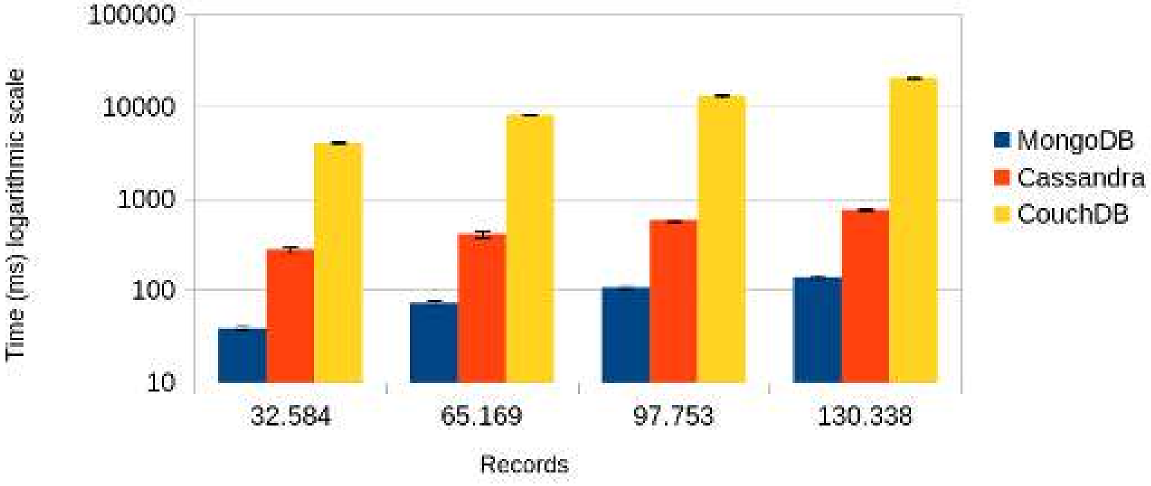


Figure 4.21: First Query - Medium Processing Time.

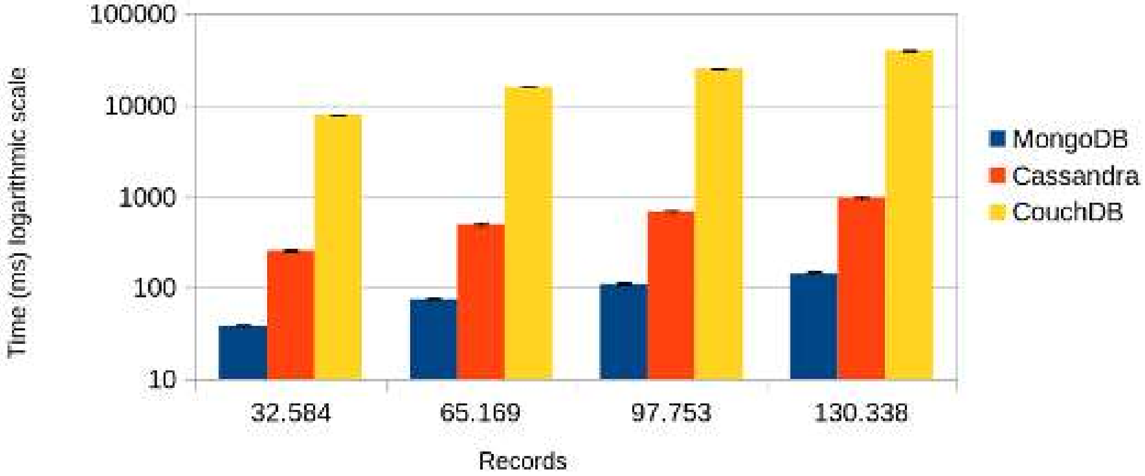


Figure 4.22: Second Query - Medium Processing Time.

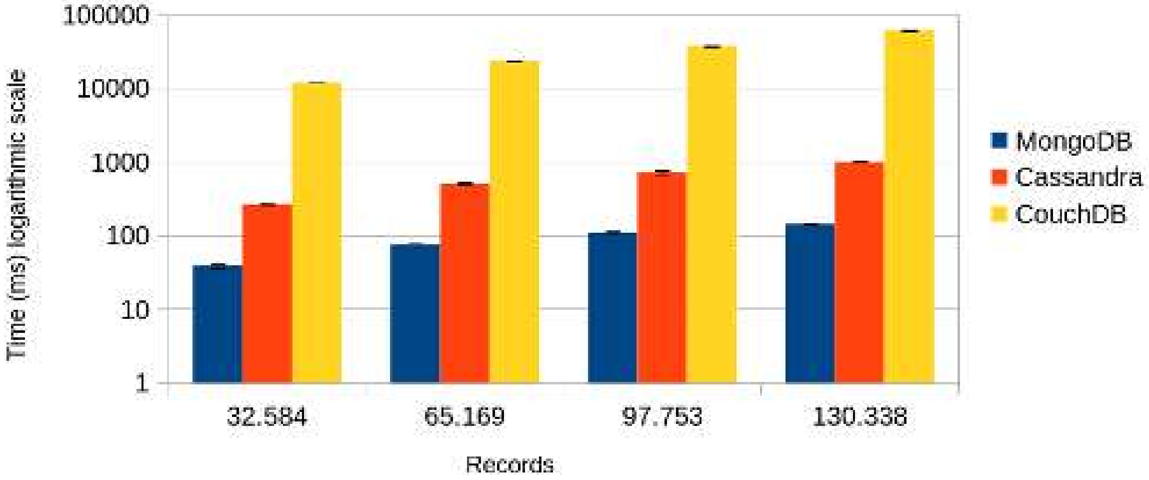


Figure 4.23: Third Query - Medium Processing Time.

that supports the described requirements (document-oriented schema-free data model, peer-to-peer architecture and TDE support) is CouchDB. The experiments, therefore, allow us to evaluate the price to pay for having a database solution fitting all the requirements of a Cloud/Edge continuum ecosystem which guarantees a high level of security through the TDE.

Table Regeneration Time

The case study considered poses the problem of data availability. Edge devices are not only prone to attack scenarios but also prone to failure. If the index table is lost on the Edge device, it is necessary to regenerate it and update the content data collection in CouchDB over the Cloud. This is necessary due to the previously described behaviour of the AES algorithm. Such a procedure makes it possible to keep the system as a whole always operational. The experiments show that the time required for encrypted data regeneration, and therefore for the recovery of the system after an attack or a hardware failure in the Edge device, grows linearly as the dataset grows. This result shows the efficiency of the implemented solution in terms of performance and therefore usability.

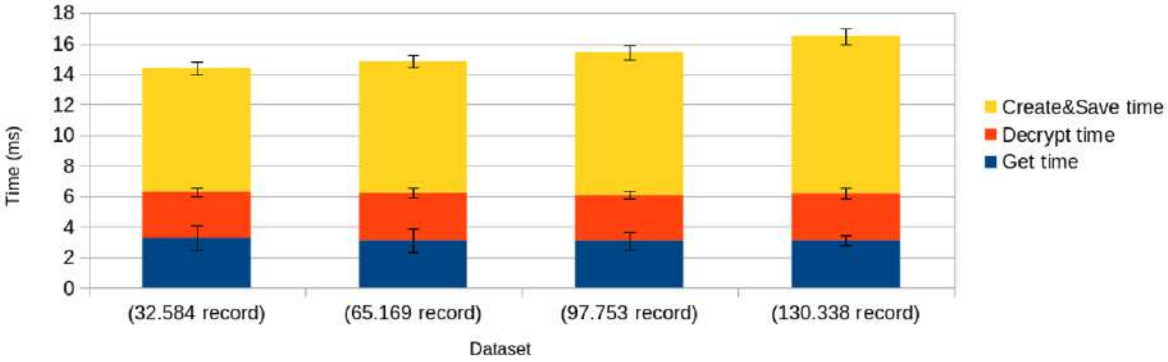


Figure 4.24: Time Keys Regeneration - Medium Processing Time.

4.2.11 Final Remarks

The work described allowed us to study, design, and develop a system that would allow us to guarantee data security in a Cloud/Edge continuum system. The developed database system meets the requirements of the document-oriented schema-free data model, peer-to-peer architecture, and TDE support. This objective has been achieved with the use of CouchDB into which the TDE has been integrated. From performed experiments emerged that the TDE solution based on CouchDB is slower in terms of computational times than the other tested solutions. However, this result is compensated by the higher degree of fault tolerance that our solution manages to give to distributed Edge/Cloud continuum systems. We think that our solution can fit different application scenarios such as healthcare and financial ones in which security and privacy are crucial aspects. In future works, we plan to optimise our solution in terms of flexibility and processing time.

CHAPTER 5

Reliability in Data Acquisition Systems

In the field of Smart Cities, the use of low-cost IoT devices is considered an advantageous solution due to their easy availability, cost reduction, and, consequently, technological and methodological development. This type of device shows many critical issues in metrological and reliability terms. This issue can significantly compromise their functionality and safety. In addition, inaccurate data constitutes low-quality input for Big Data systems. This translates into decision support systems that present solutions that are not in line with environmental reality. To better understand the degradation phenomena was interesting to evaluate the effects of accelerated aging. This was possible to study in extreme climatic conditions the performance of an IoT system based on a low-cost device. To solve these limitations a possible solution was resumed in the concept of IoT rejuvenation. This is a proactive cost-effective technique that can contrast the inevitable ageing of IoT systems guaranteeing the accuracy of collected data over time. This chapter is reporting an experimental study to demonstrate the seriousness of the IoT aging issue. To achieve such a goal, was consider a smart city scenario including intelligent street pole lamps equipped with low-cost ultrasonic sensors able to switch on lights when a vehicle is detected. Numerical analyzes were carried out to define the systematic periodicity of the errors. The purpose of the study is to lay the basis for defining a transfer function that can correct the error. Another use case to test low-cost IoT has always considered the smart city. An Edge/Cloud system has been configured for monitoring some environmental and non-environmental parameters. The device was installed on a light pole and the measurements were subsequently processed in the laboratory by evaluating them with respect to measurements obtained in a controlled environment.

5.1 IoT Rejuvenation for Sensors Reliability

The continuous development and exponential growth of smart technologies and infrastructures require adequate investments. The global market for intelligent industries and cities is estimated to reach a value of 820.7$ billion by 2025 [90]. This acceleration is mainly due to the development of Internet of Things (IoT) technologies, global regulations, and the arrangement of the 5G network. The COVID-19 pandemic has also greatly affected the production of IoT devices. Notably, some companies have seen a 50% drop in IoT spending [91]. The spread of smart applications in different sectors has raised many problems related to the costs of the sensors installed on IoT devices. In various scenarios, such as smart cities, smart agriculture, and digital healthcare, the need for large-scale devices is increasing. This means high costs in terms of IoT device manufacturing. For this reason, the use of low-cost sensors can be a solution to be adopted on a large scale in various IoT application scenarios. Nevertheless, the main problem of using such low-cost sensors lies in the uncertainty of the data they can acquire, transmit and process. A high error of the received information can cause different issues, such as a flawed risk assessment or damage to the entire ecosystem in which they are inserted, or even it can cause problems for people who use them.

The inaccurate data acquisition of low-cost sensors often makes worse constantly over the time with the intensive use of the device. We define such behavior as "sensor ageing".

Definition 5.1.1. *Sensor ageing is the tendency of the device to fail or cause a system failure after running continuously for a certain time, or because of ongoing changes in the surrounding environment.*

Therefore, sensor-equipped IoT devices must be continuously monitored in order to be properly re-calibrated if required in an automatic fashion. For this reason, a set of methods and algorithms that allow the IoT device to be maintained efficiently over time are required. To this end, inspired by the software rejuvenation theory, we propose the new concept of "IoT rejuvenation".

Definition 5.1.2. *IoT rejuvenation is a proactive cost-effective technique consisting of a set of measurement, calibration, and optimization software algorithms that enables the IoT device to properly work in a constant manner over time from the point of view of data acquisition accuracy and processing.*

The main objectives of this preliminary scientific work in the perspective of IoT rejuvenation are: *i)* demonstrating that sensor ageing is a sensitive issue that regards the whole academic and industrial communities; *ii)* proposing a reference IoT device model and related technologies.

For demonstrating sensor ageing, we consider a low-cost IoT application based on ultrasound HY-SRF05 sensors. As depicted in Figure 5.1, the sensor is installed in a smart street pole lamp that is activated when a target, i.e., either a person or a vehicle passing in the street, is detected. The ultrasonic HY-SRF05 sensor is integrated with a Micro Controller Unit (MCU) enabling it to acquire distance information in a specific time period. Each sensor has its own behavior and for this reason, it is important to understand how it works according to the surrounding environment. As shown in Figure 1, IoT devices installed on smart street pole lamps are isolated from others. This means that each single ultrasound sensor can detect only the target that is in its spatial range. A target detection error can raise different problems, such as the activation of the lamp with a consequent consumption of energy, or its total malfunction.

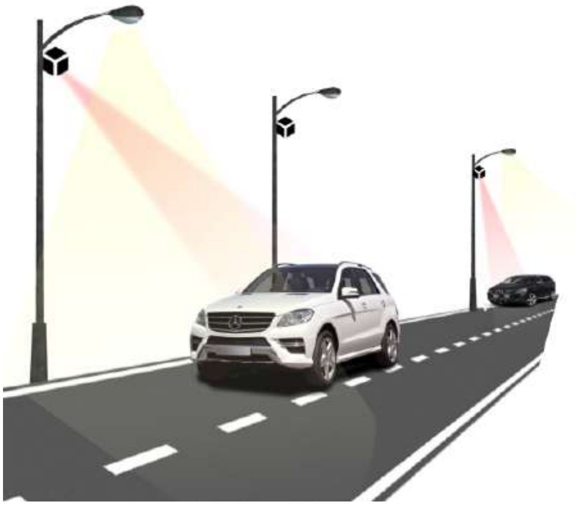


Figure 5.1: A sensor ageing scenario: smart street pole lights.

As it will be demonstrated later on in this paper, each HY-SRF05 ultrasound sensor is affected by ageing: this means that starting from a certain instant, it will begin to constantly degrade the accuracy of acquired distance information. For this reason, the MCU must be able to detect such degradation and to self-calibrate the sensor so as to make a good target detection again. The self-calibration process for the ultrasound sensor is related to the reference target and it is made in correlation with its unique transfer function, i.e., a mathematical function that theoretically models the system's output for each possible input.

Since every single sensor has different behavior and can also have various reference values according to those reported in the technical data sheets by the manufacturer. Starting from such a problem, in this paper, we also propose a reference IoT rejuvenation-enabled device model enabling us to define a dynamic calibration system based on Function as a Service (FaaS) at the Edge having a great efficiency in terms of scalability, cost, and management in the perspective of Cloud/Edge continuum [92]. The FaaS model is based on scalable and flexible events, and therefore it is an optimal choice for IoT solutions and data processing on the Edge [93]. Our reference IoT device model represents a blueprint that enables software developers to accomplish mechanisms able to systematically identify the error committed by the sensor due to ageing and to develop re-calibration strategies.

5.1.1 Related Work

In this section, we introduced the new concepts of sensor ageing and IoT rejuvenation. Since there are no specific related works on these emerging topics at the time of writing this chapter, in this Section, we provide an overview of error detection and software rejuvenation in the sensor context, highlighting the differences with our scientific work.

Sensor Error Detection and Correction

The use of low-cost sensors is advantageous but can lead to problems related to data acquisition and subsequent analysis. Different studies have been carried out, both in terms of the precision of the acquired data and of production costs. In the study proposed in [94], the authors present an Artificial Intelligence based application for plaque monitoring and animal intrusion detection. For this purpose, Raspberry Pi and several sensors are used including ultrasonic sensors. In the paper, more attention is paid to Machine Learning algorithms, but an analysis of the accuracy of the sensors is not carried out. This can lead to an acquisition system affected by errors which negatively affects the entire decision-making system. This solution risks causing false alarms that lead to sub-optimal use of the proposed IoT application. An interesting IoT application in the field of Fog computing is presented in [95]. The authors describe how to use Fog computing technology for data acquisition in a Recirculating Aquaculture System (RAS). In the proposed solution, ultrasonic sensors are used to monitor the water level in the system. A Rasberry Pi is used for processing to connect the sensors to establish a communication channel between the RAS and the Cloud over the Internet. Considering the type of sensors used and their number, it would be necessary to understand whether the numerous sensors operating together provide correct or error-corrupted measurements. In [96] the authors set up a test badge for self-driving vehicles. The application requires high precision, but an initial calibration of the sensors is performed without any algorithm that refines the measurement in real-time. The system could be built with cheaper sensors using an on the Edge system to reduce the error of low-cost sensors. The last part of the study is focused on ultrasonic sensors used in IoT applications. In [97] the authors measure statically distances (meter and stationary targets) with a low-cost ultrasound sensor by sending the data to a smartphone. The results show high accuracy in the measurement of the sensor used in the specific use case. However, concerning the proposed work, it is noted that the tests are not sufficiently exhaustive as there are not a large number of surveys but using a smartphone to view the measurement. The data reported, therefore, do not allow the validation of the measurements by affirming the goodness of the sensor. An interesting study on the use of ultrasonic sensors for the automatic movement of the robot is reported in [98]. The authors show that triangulation between two ultrasonic sensors reduces the number of sensors to be mounted on the robot. However, nothing is said about the accuracy of the measures. This means that it has to be proven that the use of triangulation reduces measurement errors by improving the movement of the robot. In [99] the authors study the measurement problems of different low-cost sensors. In particular, the study shows how several low-cost ultrasonic sensors perform measurements with errors. By comparing the characteristics of different sensors, the authors tried to figure out which is better. The paper does not report experimental data, however, it is noted that the sensor considered the best is that used in the test phase in the proposed work.

Software Rejuvenation in a Sensor Context

Software Rejuvenation has been adopted in a few works in the sensor context so far. One of the first initiatives was performed during the updating of the MACAO curvature wavefront sensor, a generic adaptive optics sensor, for a very large telescope [100]. Specifically, obsolete components were corrected and new capabilities were added. However, the rejuvenation of the project regarded meanly the hardware and were performed manually by technician without any automatic self-calibration software. The potential of software rejuvenation for long-running sensor network deployments is discussed in [101]. Specifically, it is described how software rejuvenation can be applied to resource scarce sensor nodes, which are tightly coupled distributed system. A survivability model of wireless sensor networks using software rejuvenation through Markov process is discussed in [102]. In particular, costs due to downtime during the rejuvenation process is discussed. An alternative software rejuvenation approach aimed at the reconfiguration for enhancing survivability of sensor networks is discussed in [103] The perspectives toward optimal software rejuvenation in wireless sensor networks using self-regenerative components with the purpose to increase the availability of sensor nodes are discussed in [104]. A piece of framework for enhancing the survivability of sensor networks using self-regenerative software rejuvenation and reconfiguration is discussed in [105]. Authors use the self-regenerative capabilities for detecting misbehaving in node level and apply software rejuvenation and reconfiguration methodology or both in order to extend the availability of sensor networks. In [106] the problem of software rejuvenation in the hospital sensor networks is addressed. The paper addresses the problem of availability using 3 different models studied with Petri nets.

Discussion and Differences With Our Work

The state of the art clearly shows the need to use low-cost sensors for large deployments. It emerged that studies have been conducted in various fields and methods have been found to reduce errors. However, for ultrasound sensors, the presented studies do not experimentally address the problem and do not propose a solution based on experimental analyzes. To this end, the solution we present in this paper aims at proposing a method to test the reliability of a low-cost ultrasonic sensor based on experimental tests. For this purpose, IoT devices will be used for data acquisition to be able to perform a statistical analysis on large numbers that will allow conclusions to be drawn on the goodness of the measurement carried out, laying the foundations for the study of measurement correction algorithms. Regarding software rejuvenation, it has been applied mostly for wireless sensor network optimization. In fact, currently, there are no available related works focusing on the self-calibration of the device due to sensor ageing. A preliminary study from the point of view of HY-SRF05 sensor calibration and accuracy analysis methods is reported in [6].

5.1.2 Overview on IoT Rejuvenation

IoT rejuvenation is a proactive cost-effective technique consisting of a set of measurement, calibration, and optimization algorithms that enables the IoT device to properly work in a constant manner over time from the point of view of data acquisition accuracy and processing. In this Section, we discuss its main phases and a model of enabled IoT devices.

IoT Rejuvenation Phases

The exponential growth of smart applications in different fields has raised many problems, one of which is related to the cost of sensors on IoT applications. Although going to use low-cost sensors is an optimal choice, from an economic point of view, one of the most critical aspects concerns their ability to acquire data and measurements in terms of errors and accuracy. Furthermore, the valuation of the calibration processes of low-cost IoT sensors is of fundamental importance, and very often not covered by scientific literature. As we analyzed in Section 5.1.1, many studies have focused on the performance of different low-cost sensors, their acquisition capacity, and static calibration methods, but no one has ever considered the possibility of calibrating low-cost sensors by using an IoT rejuvenation approach that allows automating the calibration processes on the Edge. On the opposite, going to install an un-calibrated sensor inside an IoT application can raise many problems, such as false data can be acquired, datasets can be enriched with untrue data, and it is possible to invalidate the automatic learning processes in the applications of artificial intelligence. Starting from the study carried out on the state of the art, it is essential to define a method that allows the periodic calibration of low-cost IoT sensors in order to validate the IoT applications in which they are installed. This is possible through the adoption of a self-calibration model on the Edge in which IoT devices are deployed. In particular, is possible to have large-scale sensor-equipped IoT devices distributed over a smart city and serving different applications. To achieve this goal, sensors must be calibrated and configured prior to their installation. This choice is made in order to extrapolate the Transfer Function of each sensor and to remove the problems related to a bad calibration process. Specifically, IoT rejuvenation requires three main phases:

1. **Low-cost sensors calibration in laboratory**. In the laboratory, we proceed with an accelerated ageing process of the sensor in extreme climatic conditions (70°C and 90% of HR). This allows you to get information on the "quality" of the low-cost sensor. Within 21 days, tests are periodically carried out on the measurement capabilities of the sensor.

2. **Sensor transfer function extraction**. The collected data are processed in order to draw the temporal trends that describe the ageing process. From the curves, it is possible to derive the transfer function.

3. **FaaS based Self-Calibration on the Edge**. The FaaS is used on-site in the application

to self-calibrate the sensors of interest. Through the edge model, every single device will be able to detect errors and anomalies due to the acquisition and re-calibrate itself according to a reference target. When data is acquired that goes beyond the threshold values, the FaaS model will be able to send an alert and request maintenance or replacement.

IoT Rejuvenation-Enabled Device Model

According to the first IoT rejuvenation phase, the behavior of the considered IoT device must be validated in the laboratory. This is necessary in order to build, in phase 2, the transfer function that is to be compared with the datasheet released by the manufacturer. In the end, such a transfer function will be used by the FaaS in order to continuously monitor and fine-tune, if necessary the IoT device

The acquisition of data by a low-cost sensor is an operation that involves a certain degree of uncertainty. To this uncertainty must be added further errors deriving from the data acquisition system. The idea behind the proposed work is to use a Micro Controller Unit (MCU) connected to the sensor which, however, does not take care of the storage or sending of data. The data will go to the serial channel of the MCU and will then be read by a microprocessor unit (MPU). After that, MPU will save (or process) the acquired data. Such a flow is described is presented in Figure 5.2 This sequence of operation must include a synchronization protocol between the two devices. Furthermore, it is of interest to our analysis, carried out in the case of ultrasound sensors, to acquire the time necessary for signal acquisition. The two consecutive readings, therefore, need to be distinguished by means of a protocol. In sequence, the MCU will write on the serial channel first the signal acquisition time in seconds and then the distance in meters. The MPU will read the values and save them only after the second reading from the serial.

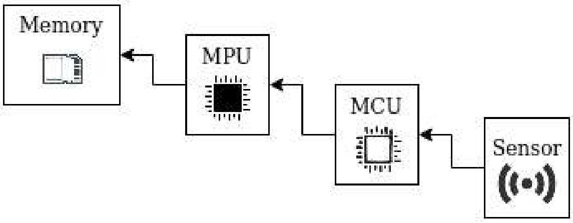


Figure 5.2: Data acquisition flow in an IoT rejuvenation enabled device model.

5.1.3 Case Study: Smart Street Lamp Scenario

We consider a smart city reference scenario including intelligent street pole lamps equipped with low-cost ultrasonic sensors able to detect the proximity of a coming vehicle to trigger when lamps must be switched on. Figure 5.3 shows how the considered smart street lamp reference scenario works. The street is the target of the model. Each single ultrasonic low-cost sensor must detect a reference distance of 4.0 meters. When an anomaly occurs in the acquired measurement, for example, 5.0 meters instead of 4 meters are detected, the sensor is self-calibrated through a FaaS process. Once the ultrasonic sensor has been auto-calibrated in relation to its transfer function, it will be able to detect the reference target by making periodic measurements. In the event that an abnormal measurement is detected, such as a distance of 2.5 meters from the target, the ultrasonic sensor will be considered unreliable and must be replaced. Through this model, it is possible to self-calibrate the sensors, without the need to have human interaction with them once installed in the reference scenarios.

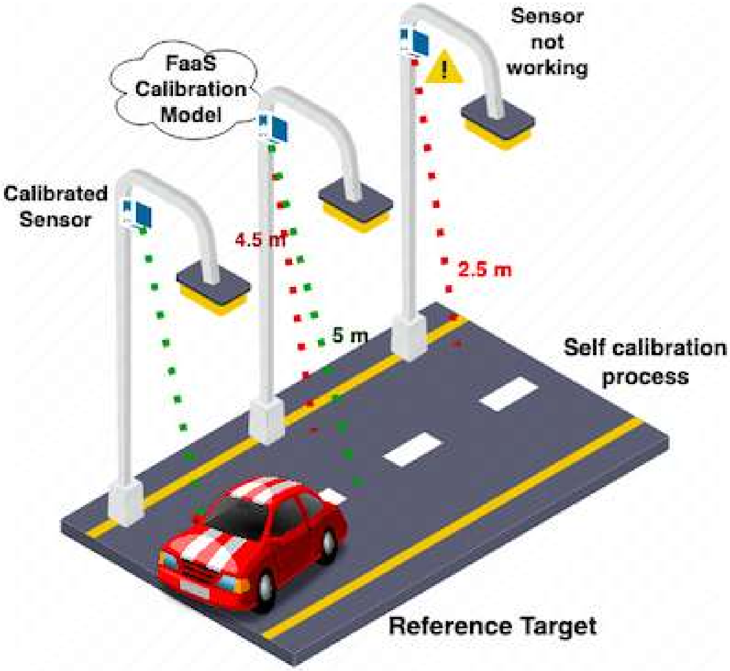


Figure 5.3: Smart street lamp reference scenario.

5.1.4 Development of an IoT Rejuvenation-Enabled Smart Street Lamp: an Overview on Enabling Technologies

In this Section, we discuss the main reference technology that can be used to develop an IoT rejuvenation-enabled smart street lamp equipped with ultrasonic device for distance detection.

Low-cost HY-SRF05 ultrasonic sensor

The investigated device (Figure 5.4) is a low-cost ultrasonic sensor (HY-SRF05), that consists of a transmitter (*T*), a receiver (*R*) and five pins. It is able to detect and measure the position of a target. Specifically, V_{c_c} and *GND* pins are used to power supply the sensor with a DC voltage of 5 V, *Trig* pin allows to send the input to start the measurement process and *Echo* pin provides the result of the measurements. Furthermore, *OUT* pin, wired to *GND* pin, ensures the 'single pin' mode. In this case, *Trig* pin can be engaged as input and output [49]. All communication pins (*Trig*, *Echo* and *OUT*) employs digital signals.

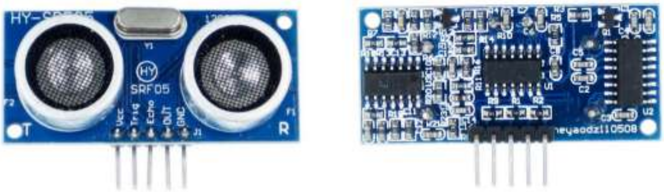


Figure 5.4: HY-SRF05 ultrasonic sensor [48], front on the left and back on the right.

HY-SRF05-target distance *d* is calculated considering the flight time *t* of the ultrasonic waves. The latter is the time necessary for the ultrasonic waves to reach the target, to be reflected and to return to *R*. This happens according to equation 1:

$$d = \frac{v_s t}{2} \tag{5.1.1}$$

where v_s is the speed of sound in dry air at 20 °C.

The ultrasound beam, generated by the sensor, does not propagate on the plane, but does so in three dimensions and in a conical way. This feature involves on the one hand a wide range of action, on the other the disadvantage of having to locate the device at a suitable height from the ground.

The scheme of the working conditions for the HY-SRF05 ultrasonic sensor is shown in figure 5.5. The measurement procedure begins when a HIGH pulse (5 V), lasting at least 10 μs [50], is sent to the *Trig* pin. This allows *T* to generate a train of eight ultrasonic pulses, each with a duration of 25 μs. Consequently, the sensor electronics sets the output of *Echo* pin at HIGH (5 V) and, simultaneously, the flight time count starts. The generated ultrasounds travel through the air, hitting the target, are reflected. When they reach the *R*, the output of *Echo* pin turns LOW (0 V) and the flight time count ends. Therefore, the flight time of the wave is equal to the HIGH period of the *Echo* signal. However, it should be noted that the generation of the ultrasonic waves is not immediate. In fact, it takes about 230 μs for the electrical charges useful to *R* to be accumulated.

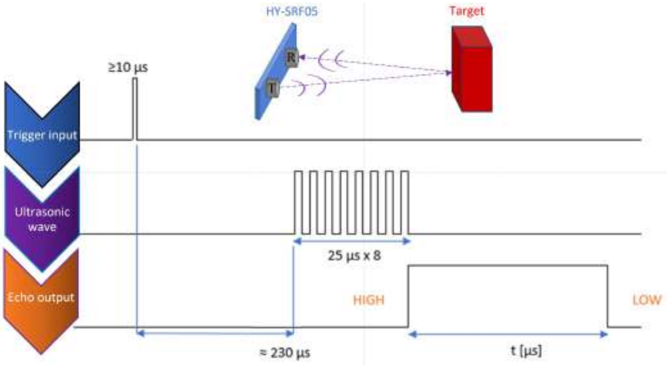


Figure 5.5: Schematic working condition for HY-SRF05 ultrasonic sensor.

Table 5.1 report the data-sheet of HY-SRF05:

Table 5.1: HY-SRF05 ultrasonic sensor specifications.

Specifications	
Working voltage	5 V (DC)
Static current	<2 mA
Input/output signals	HIGH (5 V) LOW (0 V)
Sensing angle	<= 15°
Detection distance range	2 - 450 cm
Resolution	0.3 cm
Working temperature	from $-20°C$ to $+60°C$
Weight	10 g
Sizes (l × w × t)	45 mm × 21 mm × 5 mm

Arduino Technology

Arduino Uno is an ATmega328P based microcontroller board. It has 14 I/O pins, 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header and a reset button. Through the I / O pins it allows the interaction of the MCU with different types of analogue or digital sensors. The board allows the serial communication of the MCU through the USB port with MPU devices such as Raspberry PI which in the case study is used as an acquisition and storage device. Arduino boards allow you to program the MCU with a C-like programming language.

Raspberry Technology

Raspberry Pi (RPI) is a single-board-computer, that is a board with an integrated MPU of I / O pins, USB ports, micro-sd card reader, Ethernet port and other peripherals depending on the model. connecting it to the power supply to a monitor, a keyboard and a mouse. The boards are also equipped with WI-FI connection and therefore it is possible to use them both as devices on the Edge for local processing or as a device to be used to communicate an MCU with cloud systems. In the proposed case RPI is used for on-the-edge processing. The RPI boards allow you to run scripts written in different programming languages such as Python.

5.1.5 Experiments on Sensor ageing

The objective of this Section is to demonstrate the motivation behind IoT rejuvenation, that is sensor ageing. To this end, by considering our smart street lamp scenario, we performed several experiments in the laboratory in order to demonstrate the degradation over the time of the HY-SRF05 ultrasonic sensor. Specifically, we accelerated the ageing of the HY-SRF05 in a climatic chamber and we analyzed its behavior [6]. The latter is an instrument capable of guaranteeing a fixed temperature and humidity for a certain period of time. Periodically, the sensor was extracted from the chamber so as to asses acquired data.

Setup and Test Method

The experimental setup summarized in Figure 5.6 consists of the HY-SRF05 ultrasonic sensor, a laser distance meter (mod. WDM 100, Würth), and a carriage. The latter is free to move along a straight path and, on it, the target is constrained. The sensor is power supplied and managed by Arduino Uno Rev 3, while the measurement data is acquired by Raspberry Pi 3 Model B +, wired to Arduino itself. The target is a polyurethane panel with sizes of 30 cm × 30 cm × 0.5 cm. This simulates the passage of a vehicle, a person, and even a small animal on the street. The testing procedure of accelerated aging was carried out using a climatic chamber (mod. 2500, Thunder Scientific), considering a constant temperature of (70 ± 0.1) °C and relative humidity (HR) of (90 ± 0.1)% for 21 days. Periodically, the measurements were conducted by removing the sensor from the chamber and waiting for 15 min to guarantee the ambient conditions of 25 °C ± 1 °C and 50% ± 5% of HR.

The position of the target was detected in a range between 15 cm and 500 cm from the HY-SRF05 ultrasonic sensor. The behavior of the sensor outside the range of the sensing distance was deliberately considered. Furthermore, through Arduino, the HIGH period of the trig pulse (t_{T_H}) at the sensor input was varied between 5 and 100 μs, in order to investigate the effect on the static measurements. Finally, the target-sensor distance overtime at a speed of 2.5 mm/s, when it approaches the sensor, was also evaluated.

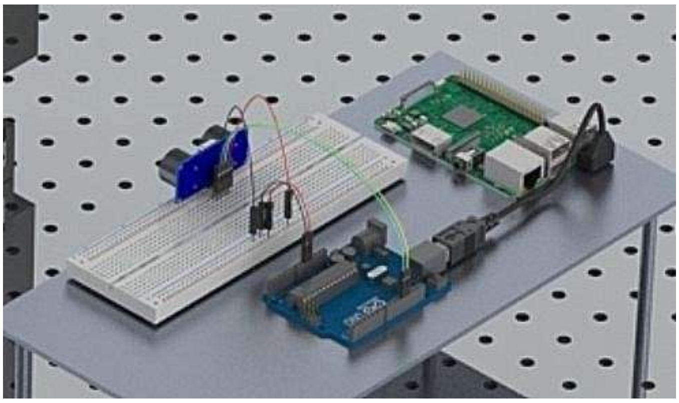


Figure 5.6: Experimental setup.

Data Acquisition

Experiments were performed in the laboratory according to the scheme presented in Figure 5.2. The components used are schematized in Figure 5.7.

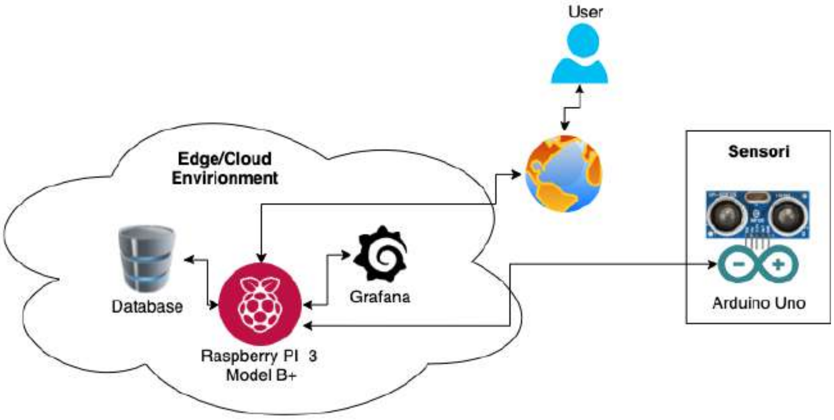


Figure 5.7: System components.

The HY-SRF05 ultrasonic sensor was connected to the Arduino board which, through an Analog-Digital converter, allowed to detection of a voltage variation which was then converted into a distance. The Arduino MCU also allowed us to detect the time required for measurement as the difference between the instant in which the impulse started and the instant in which it returned to the trigger. These values were written on the serial communication channel with the Raspberry PI board connected via a USB port. Writing took place every two operations, as per protocol, in order to acquire reading time and target distance. The Raspberry PI acquired data by updating a No-SQL Database. Moreover, thanks to the Grafana user interface it was possible for the user to understand in real-time the results of the test. This procedure permitted stress the sensor to verify the correct functionality.

Results

The performance evaluation of the HY-SRF05 ultrasonic sensor was performed by means of the investigation of both static and dynamic behaviors [6]. Figure 5.8 shows the results of the calibration procedure for the HY-SRF05 ultrasonic sensor in a static working condition. Every reported point is the mean of a signal of 10 s at a sampling rate of 50 Hz.

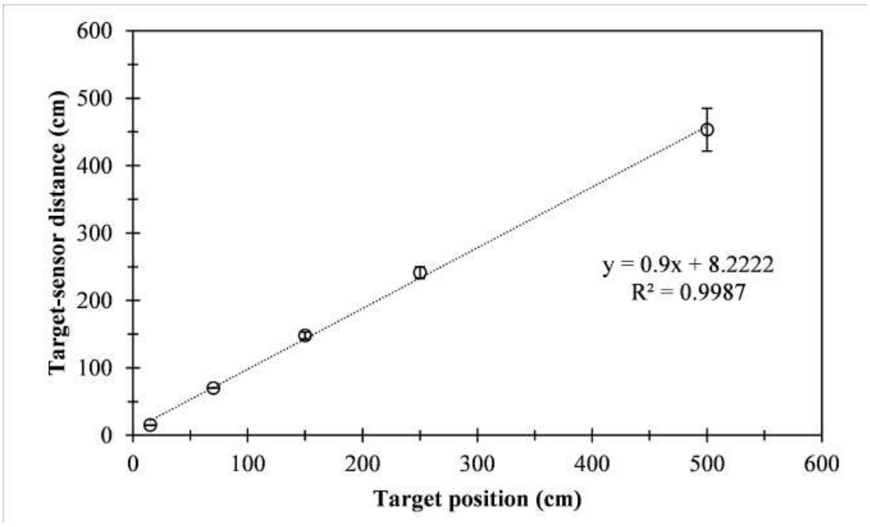


Figure 5.8: Calibration curve of the HY-SRF05 ultrasonic sensor.

In the investigated range, the calibration curve had a good linear trend with a limited and generally constant standard deviation. Only when the target was at 500 cm from the sensor, i.e. out of range of the sensing distance, did the standard deviation value rightfully become really relevant.

Figure 5.9 displays the influence of t_{T_H} on the calibration curve of the HY-SRF05 ultrasonic sensor.

Although the sensor datasheet suggests employing a t_{T_H} of 10 μs, the measurements did not highlight a predominant effect of the variation of such parameter in the range of 5-100 μs. In fact, all the reported calibration curves followed the same trend and have similar standard deviation values. Therefore, t_{T_H} did not influence the generation of the train of eight ultrasonic pulses.

Figure 5.10 completes the static investigation, putting into evidence the accuracy of the HY-SRF05 ultrasonic sensor.

The sensor accuracy is estimated in accordance to the following equation:

$$sensoraccuracy = \frac{Targetsensordistance * 100}{Targetposition} \tag{5.1.2}$$

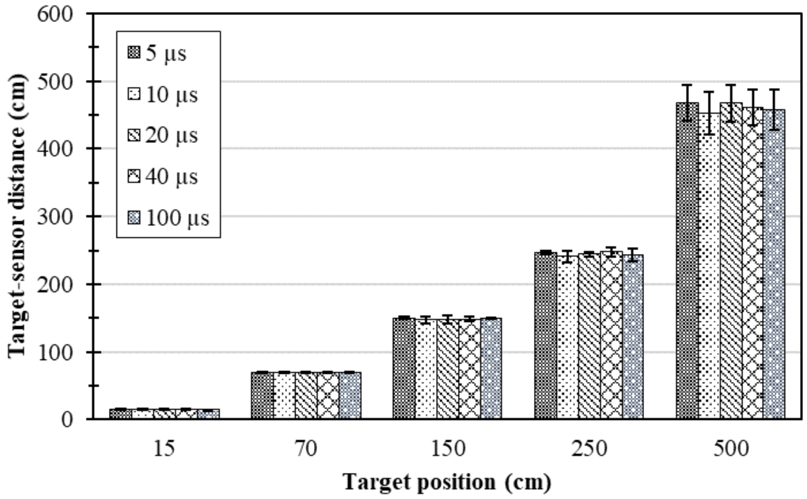


Figure 5.9: t_{T_H} influence on the measuring performance of the HY-SRF05 ultrasonic sensor.

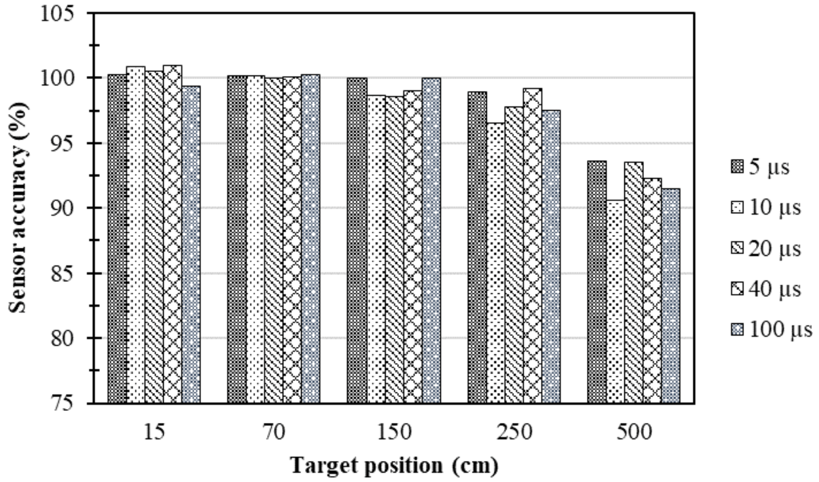


Figure 5.10: Accuracy of the HY-SRF05 ultrasonic sensor, considering t_{T_H} variation.

Where Targetsensordistance represents the distance between the target and the sensor measured by the sensor. Targetsensor instead represents the distance between the target and the sensor measured with a laser meter. This measure was validated with instruments with a limited margin of error and is therefore an element of comparison. Accurancy seemed sufficiently constant in the range of the sensing distance, but after that, i.e. at 500 cm, a marked reduction was observed. Generally speaking, the sensor accuracy represented a relevant disadvantage because it caused an important noise in the light control. It should be noted that we find values over 100% because the value acquired by the sensor can also exceed the maximum reported by the manufacturer. This situation can depend on IoT malfunctions in a particular condition or can often be outliers.

Figure 5.11 exhibits an example of the dynamic behavior of the HY-SRF05 ultrasonic sensor. In this case, the target approached the sensor with a constant speed of 2.5 mm/s and the signal from the sensor was acquired for 60 s at a sampling rate of 50 Hz.

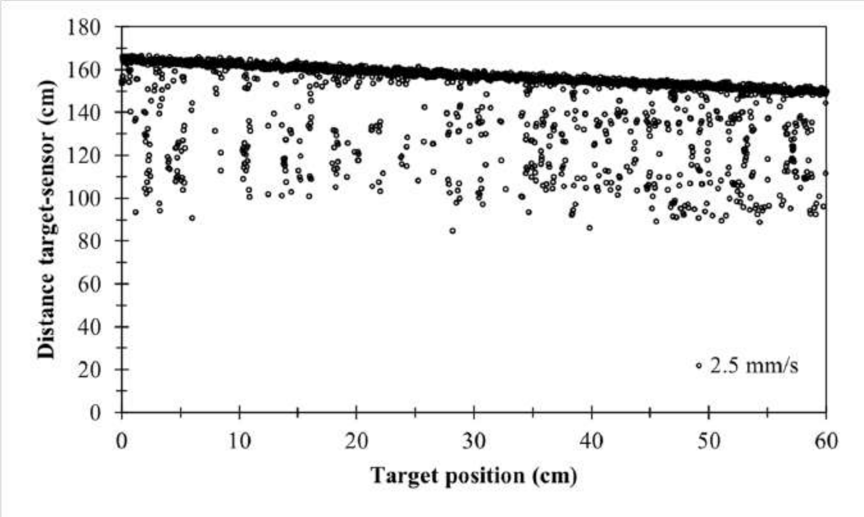


Figure 5.11: Dynamic measurement of the HY-SRF05 ultrasonic sensor.

Although the trend was sufficiently linear, many outliers were present. These can compromise the correct working operation of a smart street lighting system, in which a standalone control should be considered unstable. To overcome this limit, it could be useful to adopt an algorithm that filters the measured signal or to insert a delay on the lighting turn-off.

Figure 5.12 reports the preliminary results regarding the accelerated aging of the HY-SRF05. This behavior was evaluated considering the position of the target equal to 250 cm from the sensor.

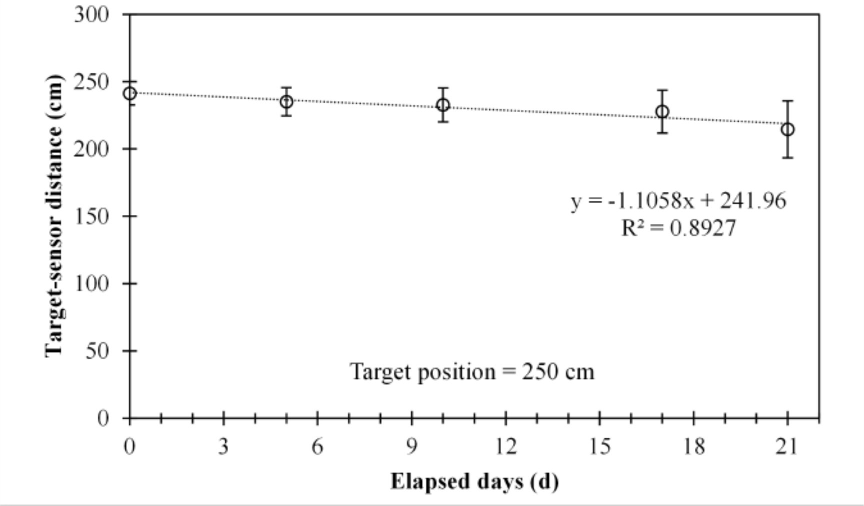


Figure 5.12: Dynamic measurement of the HY-SRF05 ultrasonic sensor

As the days of treatment at high temperature and humidity pass, the sensor exhibits a gradual reduction in its performance. Although the trend can be considered of linear type, at least as a first approximation, the measurements are also subject to a widely increasing of the standard deviation over time. This has fundamental importance since it determines a high variability and uncertainty in defining the correct target position. A calibration procedure is, therefore, necessary to guarantee the good functionality of the controlled pole.

5.1.6 Final Remarks

Nowadays, in response to the need for large-scale deployment of IoT devices, a possible solution is represented by the adoption of low-cost sensors. Nevertheless, the price to pay is the degradation of its accuracy in collecting data over time. In fact, it emerged that the low-cost sensors often do not respect the manufacturers' specifications and above all, after a certain time of use, they are no longer allowed to acquire data correctly. The purpose of this scientific work was to point out the attention of both academic and industrial communities on a new research line, that is IoT rejuvenation and specifically, to demonstrate the motivation behind it is sensor ageing. Specifically, we considered smart street poles equipped with HY-SRF05 ultrasonic sensors as a reference scenario and we concentrated our attention on device ageing. Several tests were carried out in the laboratory trying to simulate the operating conditions in extreme situations. Recalling the reference standards, we tried to accelerate the time to systematically study the response reference IoT device. The most relevant test results have been graphed and commented on precisely in order to detect systematic errors during the ageing process of the device. As a future work, by considering the reference scenario described in this paper, we plan to mathematically extract a transfer function that will allow us to understand how the sensor can be re-calibrated in case of ageing. In addition, starting from the transfer function, we plan to design and implement through a FaaS approach an algorithm able to continuously monitor the ageing of the IoT device and fine-tune if necessary the sensor in order to rejuvenate it. Furthermore, the next activities foresee testing the sensor in a real environment by evaluating the weather effect on the sensor and its characteristics. With this paper, we hope we succeeded in opening a new research line, i.e., IoT rejuvenation and to stimulate the interest of both academic and industrial communities toward this new emerging topic.

5.2 Low-Cost Device in Urban Environment

Street lighting is an indispensable and crucial system in urban and suburban frameworks. Its main purpose is to prolong the activities of a city in the dark hours, illuminating places and streets traveled by pedestrians and vehicles. However, this involves significant costs; it requires about 13–14% of the annual production of global electricity [107] . In the context of Smart Cities, street lighting is not only reduced to an energy intensive system to be optimized, but its infrastructure can be easily employed to understand and to improve vital urban parameters. This can be achieved not only by applying new generation strategies, but also by intervening on current systems to make them innovative. The use of a sensor network and some specific communication technologies can transform street lighting poles, which are also approximately indicated as columns, into intelligent and multifunctional structures [108]. For example, it is possible to monitor environmental parameters [109], to manage vehicular traffic [11] or to implement the safety policies of a city [110, 12]. These necessities have played prominent roles in society, and a large part of recent scientific and industrial efforts are directed in such a way.

A relevant aspect, which is especially underestimated in the less frequented and more peripheral areas of a city, is the decrease in structural safety over time. A more widespread, real-time and low-cost monitoring represents a possibility of considerable interest [111].

Although an electrical malfunction [112] is really frequent, the structural collapse of street lighting poles [113] is often unpredictable and particularly destructive. Traditionally, these supports were manufactured with wood [114] due to the abundant availability and low cost of such a material. However, since they consist of an organic material, they are therefore subject to a rapid deterioration, which is mainly caused by atmospheric agents. To cope with this adversity, they were replaced with concrete, cast iron or stainless-steel columns [115, 116, 117], which are more durable but also heavier, difficult to install and expensive. One applied alternative was structural steel [118]. Unfortunately, it is prone to corrosion and requires an anti-rust coating, which increases its maintenance costs. More recently, aluminum poles have been adopted [119], but fiber-reinforced polymers are currently preferred as manufacturing materials [120]. In fact, the latter are able to combine good mechanical properties, great durability and ease of installation. The poles are designed according to some specific regulations that take into consideration several factors, including their weight, the weight of their accessories, the acting forces of wind and snow, etc. In such a case, the main cause of degradation is chemical corrosion caused by atmospheric agents, pollutants, the composition of the installation soil and galvanic couplings, which are present in different urban contexts [121]. Specifically, the part of a structure that is most subject to this degenerative phenomenon is the one in correspondence with the joint with the ground, where there is the maximum static load. It is summarily hidden underground or concealed by protective sheaths and a concrete base. For this reason, the prediction of possible structural damage becomes extensively complex [122]. However, a structural collapse can also occur due to extreme causes, such as road accidents [123] and adverse weather conditions [124]. In any case, the collapse due to structural fatigue induced by normal atmospheric conditions is configured as primary and, above all, sudden [125]. Taking into account the previously described sources of damage, the residual life of a generic street lighting pole is a complex parameter to assess and is subject to various evaluation errors [126]. Typically, it is estimated by computing the corrosion rate and measuring the residual thickness of the structure or by adopting a nondestructive method [127]. Few researchers have proposed different alternatives. For example, Ziolkowski et al. [128] analyzed the response of some artificial defects in poles by means of short-range ultrasonic guided wave technology and time–frequency decomposition and conceptualized this procedure for an executive application. Although such techniques remain valid locally and, above all, in specific cases, the large number of installed columns discourages their applicability on a massive scale due to the unsustainability of noncontact tests and enormous costs [129]. Currently, the most employed method remains manual or semi-automatic visual inspection unit by unit [130].

An attractive alternative is to perform an SHM (Structural Health Monitoring) strategy employing a dedicated system integrated in single street lighting poles and in accordance with the typical exigencies of a Smart City [6]. Currently, this remains a field of interest that is still limited both in the literature and on the market. Steinbauer et al. [131] have recently developed a method based on the determination of the natural frequency shift through environmental excitation from wind or traffic. Specifically, they employed a device consisting of a MEMS (Microelectromechanical System) accelerometer glued to the top of a pole. It was connected to a wireless network grid that was capable of sending the obtained measurements to a Cloud system and subsequently achieving the results from a prototype software developed in a Matlab/Simulink environment. Reverberi Enetec srl [132] introduced a triaxial accelerometer and a triaxial inclinometer in the nodes of its commercial remote control systems. The collected information is transferred via a proprietary Internet infrastructure to another proprietary remote management software for a subsequent analysis. Through these data, it is possible to monitor the oscillation and tilt of public lighting poles and, in the event of anomalous variations (collisions or structural failures), to send an automatic alert to an emergency response team. The main disadvantage of this system concerns its intrinsic commercial nature, which involves the exploitation of specific sensors and proprietary software and infrastructure.

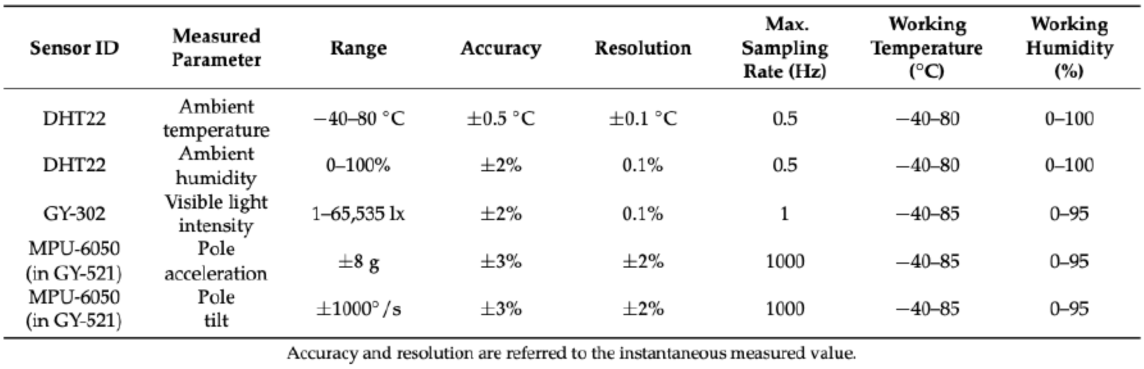
The presented paper reports the design, the implementation and the experimental characterization of a low-cost system for monitoring the structural behavior of street lighting poles in Smart Cities. It consists of an acquisition and transmission device wired to a specific set of sensors and connected to an Internet network. The aim of this work is to introduce a multi-functional device that can be easily integrated on an existent urban and suburban framework. It has the ability to collect the previous structural information of the poles, to simultaneously measure meteorological aspects and to lay the foundations for the development of a method that is able to avoid the collapse of the same poles. The monitoring architecture is presented in the aspects concerning the software programming, the management and the visualization strategy of the acquired data and the sending of specific alerts to the control room. The described experimentation has been focused on the estimation of the metrological performance of the proposed system.

5.2.1 Materials and Methods

Low-Cost Monitoring System

The proposed system was a low-cost dedicated hardware consisting of an acquisition and transmission device wired to a set of sensors. It was inserted in a commercial case in plastic material with an IP 56 protection rating for outdoor use. To the latter, a circular support was screwed to its bottom to fix it to a street lighting pole.

The device was a single-board computer (Raspberry Pi 3 B+) supplied by an electrical grid via an on–off switch and equipped with two LEDs (Light-Emitting Diodes) in order to indicate its working status (green LED = powered; red LED = not powered). It was employed to acquire information, to locally store and to subsequently send the collected information via mobile connection. The sensors Figure 5.13 had the aim of identifying possible structural instabilities or failures of the pole due to dynamic human and environmental effects, and to simultaneously monitor some meteorological parameters. Specifically, the ambient temperature and ambient humidity were measured using a DHT22 sensor [133]. It was fixed on a thermally insulating support on one side of the case and protected by a Stevenson screen in ASA (Acrylonitrile Styrene Acrylate), which is also suitable for outdoor use and is UV (UltraViolet) resistant. The visible light intensity (wavelength 400–700 nm) of the solar radiation was evaluated using a GY-302 sensor [134], which was installed on the top of the case and covered by an opalized glass bulb. This sensor was capable of guaranteeing the transmission of solar rays and the insulation from atmospheric agents. The vibration and tilt of the pole, due to environmental causes and the passage of vehicles, were instead assessed using a GY-521 module equipped with an MPU-6050 sensor [135] placed inside the case integrally. All sensors were directly powered by the single-board computer. The GY-521 module was adopted by considering the data reported in [131] and the typical dynamic behavior of a street lighting pole. This aspect, which is well consolidated in the literature [136], is of considerable relevance as it depends on numerous intrinsic factors of the structure (e.g., material, geometry, and weight) and external agents (ambient deterioration and damage caused by third parties).



Sensor ID	Measured Parameter	Range	Accuracy	Resolution	Max. Sampling Rate (Hz)	Working Temperature (°C)	Working Humidity (%)
DHT22	Ambient temperature	−40–80 °C	±0.5 °C	±0.1 °C	0.5	−40–80	0–100
DHT22	Ambient humidity	0–100%	±2%	0.1%	0.5	−40–80	0–100
GY-302	Visible light intensity	1–65,535 lx	±2%	0.1%	1	−40–85	0–95
MPU-6050 (in GY-521)	Pole acceleration	±8 g	±3%	±2%	1000	−40–85	0–95
MPU-6050 (in GY-521)	Pole tilt	±1000°/s	±3%	±2%	1000	−40–85	0–95

Accuracy and resolution are referred to the instantaneous measured value.

Figure 5.13: Main features of the set of sensors.

Finally, a small fan, supplied by a USB socket wired to an electrical grid, was installed to control the temperature and to expel the hot air through two vents on opposite sides of the case by the device. Figure 5.14 reports a representative schema of the electrical and electronic components.

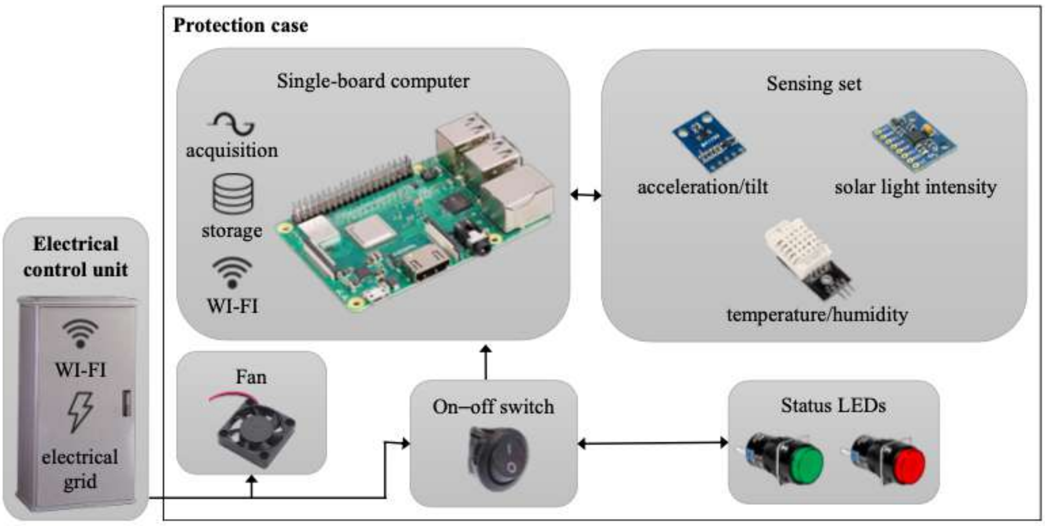


Figure 5.14: Representative schema of the electrical and electronical components of the monitoring system.

The proposed system had a volume of 190 X 140 X 70 mm3 and a weight of about 0.7 kg. It was installed on the top of a street lighting pole (Figure 5.15). The latter, about 10 m high, was chosen because it is located on a suburban road that is mainly frequented during the winter and summer holidays and is near a sports facility. Although the mechanical influence of the proposed system was not verified, the typical head of the pole has a volume and a weight 7–8 times superior. The structural resistance of the whole pole is ensured by its column [137]. In fact, although its accessories (e.g., head) are relevant to its mechanical behavior, their effect is often limited. Moreover, a possible relamping must also be considered; in this case, the new lantern has a considerably lower weight than the old original one [138]. Specifically, the power supply was taken from an electrical grid to ensure there was a significant reduction in maintenance costs thanks to the unnecessary battery replacement. Instead, the communication to an Internet network was guaranteed via mobile 4G connectivity. In any case, the proposed solution can work, as it is associated with multiple devices with a single Internet entry point, which is also wired. The described situation constitutes the typical context of interest for the application presented in such a paper. Finally, taking advantage of the fact that it is based on a Raspberry computer and has a modular configuration, the proposed system can be easily integrated into the current lighting infrastructure and enriched with other functionalities. As an example, a further investigation procedure can be implemented [139].

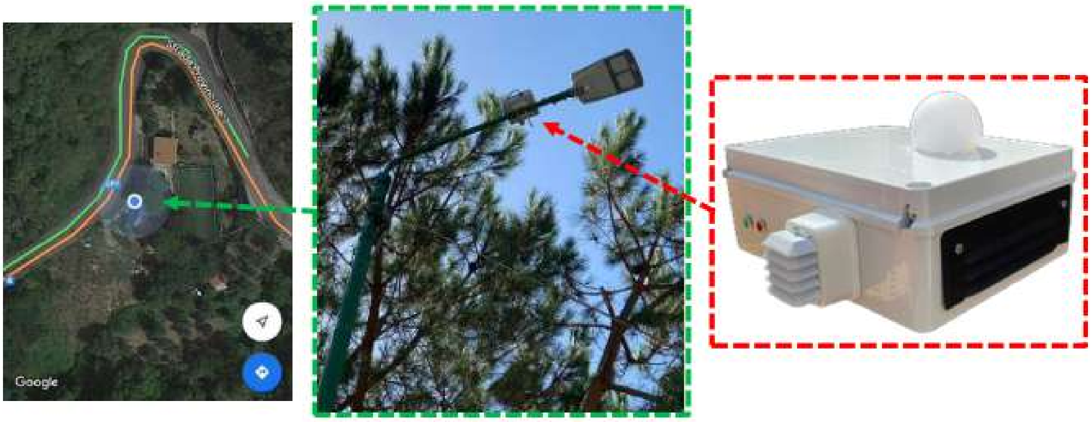
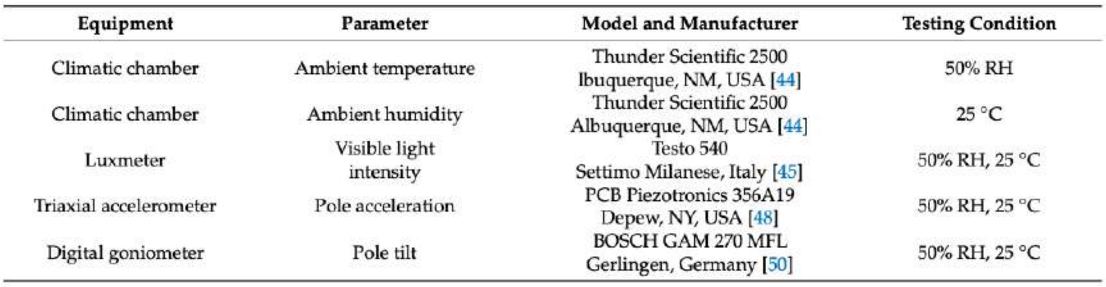


Figure 5.15: Installation and details of the monitoring system.

Characterization Setup

The proposed system was characterized by means of a different experimental setup Figure 5.17 for each specific function of the employed set of sensors. A climatic chamber [140] was employed to recreate the typical ambient conditions of the place where the proposed system was installed (Figure 3a). Specifically, as standard conditions, a constant Relative Humidity (RH) of 50% was chosen, and the temperature varied. Conversely, a temperature of 25 °C was set, and the RH changed. The visible light intensity was estimated in a darkroom (Figure 3b) at 50% RH and 25 °C as standard conditions. Inside, a halogen lamp with a maximum power of 1 kW, simulating the solar light, was used to project a controlled light beam onto the proposed system. The data were acquired and subsequently compared with those detected using a luxmeter [141]. An electrodynamic shaker (TIRA vib S 503, Schalkau, Germany [142]) supplied by a power amplifier (TIRA vib BAA 60, Schalkau, Germany [142]) and driven by a function generator (Agilent 33220A, Santa Clara, CA, USA [143]) was used to reproduce a sinusoidal dynamic stress (Figure 3c) at 50% RH and 25 °C. Specifically, a triaxial accelerometer [144] wired to an acquisition system (National Instrument cDAQ 3320, Austin, TX, USA [145]) was installed together with the GY-521 module of the proposed system on a rigid base (in aluminum with a thickness of 7 mm) directly fixed to the stinger (in steel with a diameter of 5 mm) of the same shaker. Finally, at 50% RH and 25 °C, a digital goniometer [146] was chosen to calibrate the GY-521 module, positioning the latter at a different tilt (Figure 3d), as reported in [147].



Equipment	Parameter	Model and Manufacturer	Testing Condition
Climatic chamber	Ambient temperature	Thunder Scientific 2500 Ibuquerque, NM, USA [44]	50% RH
Climatic chamber	Ambient humidity	Thunder Scientific 2500 Albuquerque, NM, USA [44]	25 °C
Luxmeter	Visible light intensity	Testo 540 Settimo Milanese, Italy [45]	50% RH, 25 °C
Triaxial accelerometer	Pole acceleration	PCB Piezotronics 356A19 Depew, NY, USA [48]	50% RH, 25 °C
Digital goniometer	Pole tilt	BOSCH GAM 270 MFL Gerlingen, Germany [50]	50% RH, 25 °C

Figure 5.16: Details of the characterization setup.

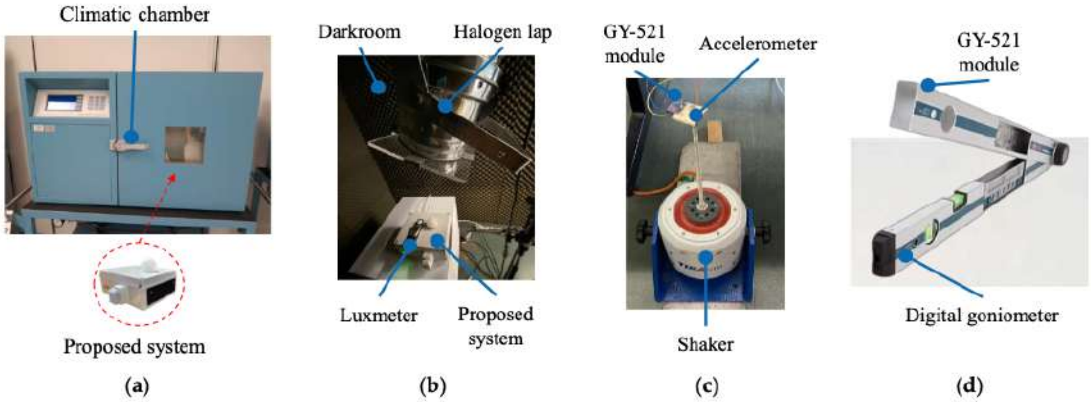


Figure 5.17: Characterization setup for the sensors of (a) ambient temperature and humidity, (b) visible light intensity, (c) acceleration and (d) tilt of the pole.

Cost Discussion

The adjective "low cost" was adopted because the proposed system is based on low-cost equipment and open-source software. These factors intrinsically ensure that the system has a limited economic effort compared to commercial devices. Indeed, although it mainly depends on the volume of its production, the estimated cost of the complete proposed system is less than EUR 200. Specifically, the used sensors (DHT22, GY-302 and MPU-6050) represent less than 1/4 of the total cost, the electrical and mechanical small parts (cables, screws, circular support, protection case, etc.) and the auxiliary devices (fan, LED, etc.) represent just over 1/4 of the total cost and, finally, the single-board computer and its accessories represent about 1/2 of the total cost. Furthermore, as already reported, it should be considered that this system is not exclusive to the discussed application and can be implemented with further functions and other devices.

5.2.2 Results

Design of the IT Architecture

The information technology (IT) architecture was based on a distributed Edge/IoT - Cloud system, which allowed the user to view, collect and manage the monitoring parameters, receiving dedicated notifications. Figure 5.18 depicts the architecture, which consisted of the following three main elements:

- *Edge Layer*.This defined the way to collect and to process the acquired data, to extract specific information and to monitor the results in real time. In addition, it was characterized by a higher level of complexity than the other elements of the monitoring architecture. In fact, it adopted a stratification and involved both parallel and confluent functions. In this work, the Edge structure consisted of the Sensor Layer, referring to the set of employed sensors (Sensor 1, . . . Sensor N). Their number and type depended on the GPIO (General Purpose Input Output) of the Edge device and on the necessary balance between their power consumption and the maximum electrical power provided by the device itself. A related service (Service 1, . . . Service N) was defined for each sensor. These were the software modules that read the data from the sensors and wrote them into a database using some dedicated drivers. For such a reason, they were strictly linked to the type of sensor and the DBMS (Database Management System) with which they interacted. Thus, the envisaged database allowed for the local storage

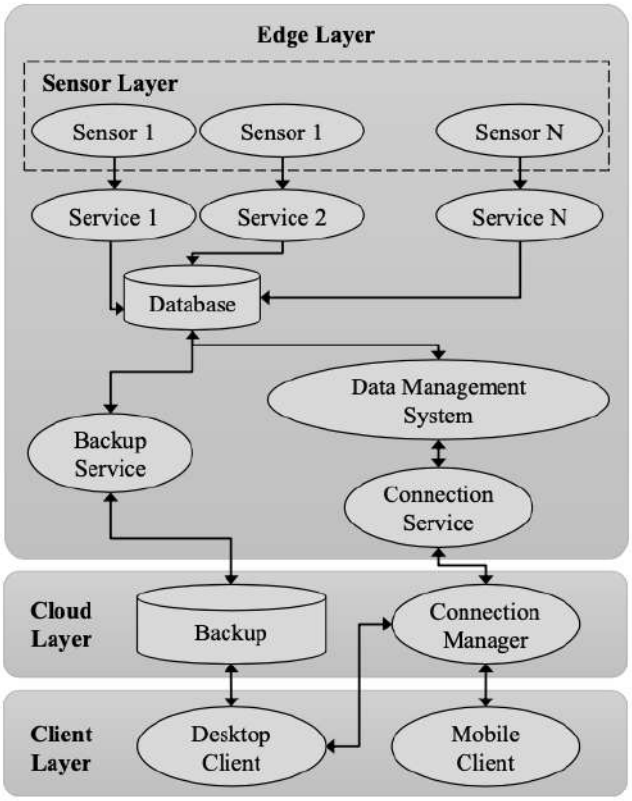


Figure 5.18: Schematic representation of the IT architecture.

of the acquired data using a Time Series DBMS. Two services simultaneously acted on the database. The first one was the Backup Service. It was a software module that periodically and automatically started a local backup of the database, transferred it to the Cloud and deleted it so as not to fill the device's memory.

The second one was the Data Management Service. This was a software module that allowed for the user to access to the database, to create customed dashboards for data visualization and to configure and send alert systems according to some specifications on the measured parameters. All of these last operations were guaranteed by the Connection Service, which consented to the user's communication with the Edge device via the Cloud Layer. Such a type of service was chosen to control the user's access to the device and to guarantee the security of the transferred data.

- *Cloud Layer*. This illustrated the elements that were necessary to remotely store and manage the collected data. Specifically, the Backup, i.e., a local database, was the component that periodically obtained updates from the Edge device via the Internet and kept a copy of these data. Moreover, it ran a software module (Connection Manager), which guaranteed the connection between the user and the Edge Layer. This connection was obtained via a VPN (Virtual Private Network) server. In this way, once the certificates were issued, the user could interrogate the Edge Layer by configuring their client.

- *Client Layer*. This described the apparatus and the software that was directly usable by the user to interact with the Edge device. The client application could be executed from both the desktop and mobile terminals, but with some substantially different purposes. A Desktop Client employed personal computers, laptops, etc., to have access to the dashboard and create a data backup via the browser and remote connection software. Instead, a Mobile Client with smartphones, tablets, etc., could contact the dashboard via a browser. However, with this limitation, such a method had the advantage of receiving alerts more immediately. For both cases, notifications could be received via email or messaging apps using specific bots.

5.2.3 Development of the IT Architecture

Figure 5.19 shows the main ITs employed for the development of each of the previously described layers Figure 5.18.

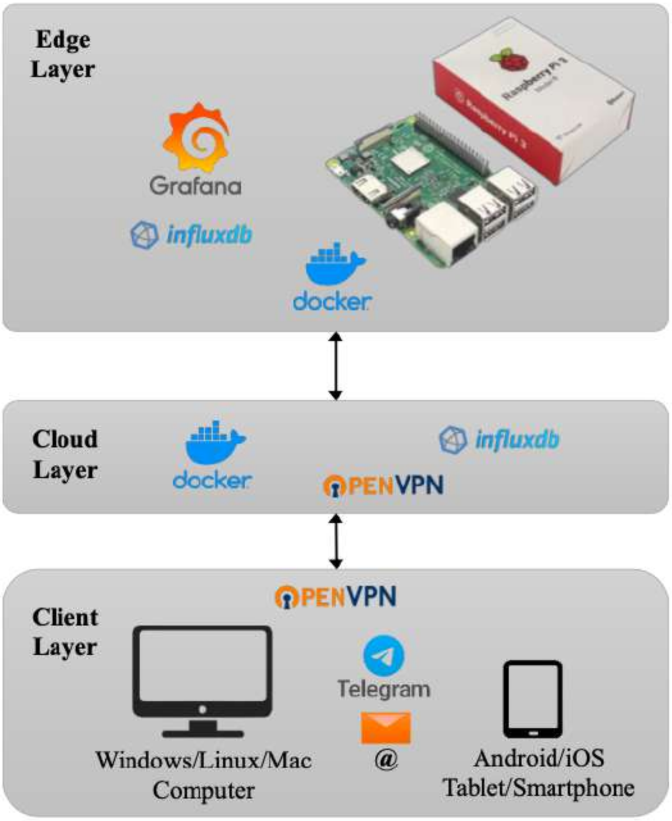


Figure 5.19: Schematic representation of the deployed scenario.

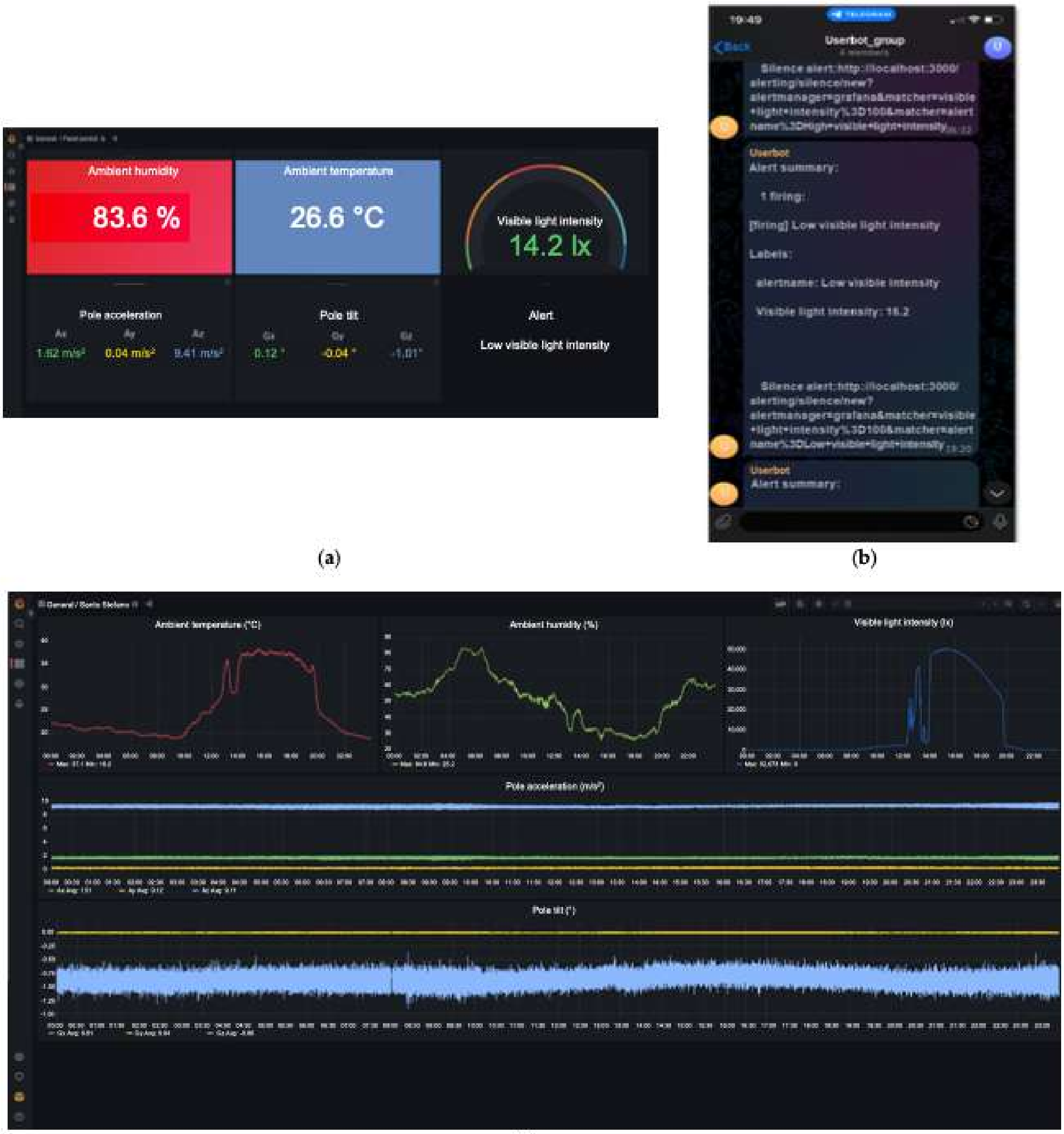
The Edge Layer adopted the single-board computer (Raspberry Pi 3 B+) as an Edge device. It was wired to the employed set of sensors (i.e., Sensor Layer) via its GPIO and equipped

with the native Raspian OS (Operating System). The collected data were saved on a database (i.e., Database), built with InfluxDB, by means of some dedicated scripts (i.e., Service 1, . . . Service N) written using the Python programming language, and managed using the Grafana web application (i.e., Database Management System). Furthermore, through the Linux OS commands, the automatic execution of the backup script (i.e., Backup Service), and the connection first to a WI-FI network and subsequently to a VPN network (i.e., Connection Service) for sending the previously saved data were set. InfluxDB, Grafana and the associated scripts were installed on the Edge device, using the container method of the Docker technology.
The Cloud Layer also exploited Docker for its services. Specifically, an InfluxDB instance performed a daily update of the backup (i.e., Backup) with all the data from the previous day, while a container instance of the server of the Open VPN software was used to issue certificates and allow for the user to connect to the Edge device (i.e., Connection Manager).
The Client Layer adopted desktop/laptop terminals with Windows/Mac/Linux OS (i.e., Desktop Client) or mobile ones with Android OS/iOS (i.e., Mobile Client), which accessed the Cloud Layer and the device via the OpenVPN and any browser. The respective features were performed via Grafana. Specifically, Grafana not only allowed for the access, the customization of the dashboard and the creation of specific data backups, but also allowed for the implementation through bots of dedicated alerts. These notifications, produced when the data reached defined thresholds, were sent via the messaging service of the Telegram app and also via e-mails, which can be received with any e-mail client.

Preliminary Field Test

Figure 5.20 illustrates the typical interfaces of the deployed scenario during a preliminary field test.

The instantaneous values of the measured parameters and an alert are indicated in Figure 5.20 a), while the related graphs are displayed in Figure 5.20 b). This alert proved the status of the "low visible light intensity" that was defined using a threshold of 100 lx. Instead, the time history of the acquired data was set at 24 h. As already stated in the previous section, according to the specific authorizations guaranteed by the suitable VPN certificates, the user is able to view a collected point or edit the different parts of the interface. As an example, the user can rescale the axes of each graph or choose to download a specific portion of the collected data. Finally, a screenshot of the display of a smartphone, on which an alert summary is provided by the Telegram app, is exhibited in Figure 5.20 c). A sampling frequency of 1 Hz was employed for the measurements of the ambient temperature, ambient



(a)

19:49
Userbot_group

Silence alert:http://localhost:3000/alerting/silence/new?alertmanager=grafana&matcher=visible+light+intensity%3D100&matcher=alertname%3DHigh+visible+light+intensity

Userbot
Alert summary:

1 firing:

[firing] Low visible light intensity

Labels:

alertname: Low visible intensity

Visible light intensity: 15.2

Silence alert:http://localhost:3000/alerting/silence/new?alertmanager=grafana&matcher=visible+light+intensity%3D100&matcher=alertname%3DLow+visible+light+intensity 19:20

Userbot
Alert summary:

(b)

(c)

Figure 5.20: Typical interface, (a) indicators and (c) graphs customed by the uses on Grafana for desktop terminals, and typical alert for smartphone on Telegram app (b).

humidity and visible light intensity. This was chosen because, reasonably, the monitored meteorological parameters vary slowly during the day. Instead, after considering exceptional events (as an example, strong windy conditions), the sampling frequency was set to 20 Hz for the acceleration and tilt of the pole. These values were only selected as preliminary ones. The collected data show that the proposed system works in a suitable way, taking into account the environmental conditions in the place where the pole is installed.

The trend of the ambient temperature is low during the night and the first part of the morning (22:00–9:30), increases during the second part of the morning (9:30–13:00), is high during the afternoon (13:00–19:30) and finally decreases during the evening (19:30–22:00). The ambient

humidity has a more complicated drift, with a progressive rise starting from sunset and a reduction starting from sunrise.
The visible light intensity displays a typical trend, but some considerations need to be made. The investigated pole is in the shade until 12:00 and thereafter is exposed to direct sunlight. Furthermore, some trees are close to the pole, and therefore, there are some important influences on the direct light. With regard to the acceleration of the pole, specific events are not clearly noted. This is due to the used time scale (24 h), which is unsuitable for short events, and due to the amplitude, which is capable of including the three components x, y and z. However, specific events can be identified during the early morning hours (8:00–8:30). These may be caused by the passage of some vehicles or by short gusts of wind. What is more interesting is the case of the tilt, where the angle along the vertical z direction (azure curve) highlights a possible and very slight relaxation of the pole during the hottest hours (12:00–19:30) and during the passage of some vehicles or short gusts of wind (8:00–8:30).

Metrological Performance

A metrological characterization was performed to investigate the possibility of implementing an accurate measurement system to monitor the structural behavior of street lighting poles in Smart Cities using low-cost devices.
Figure 5.21 reports the calibration curves and the comparison between the trends of the uncertainties, computed by the standard deviation of the collected measurements and by the declared accuracy Figure 5.13 of the proposed system with reference to the ambient temperature, ambient humidity and visible light intensity.

For the DHT22 (Figure 5.21 a,c) and GY-302 (Figure 5.21 e) sensors, each point of the calibration curves was obtained by considering the average value of three different measurements. The latter were calculated by averaging the last 10 points that were acquired 300 s after reaching stationary conditions at a sampling frequency of 10 Hz. These procedures were employed to obtain the values that best represent the performance of the investigated sensors. In fact, an average value of the specific information should be preferred in order to avoid potential outliers. The uncertainty budget was carried out in accordance with [147].
Specifically, the uncertainty relating to the collected measurements was estimated by dividing the value of the punctual standard deviation by $\sqrt{n}$, where n is the number of repetitions of the same type of measurement (i.e., 3). Instead, the uncertainty related to the data stated by the manufacturer was calculated by considering a rectangular distribution and therefore dividing the declared accuracy by 3.

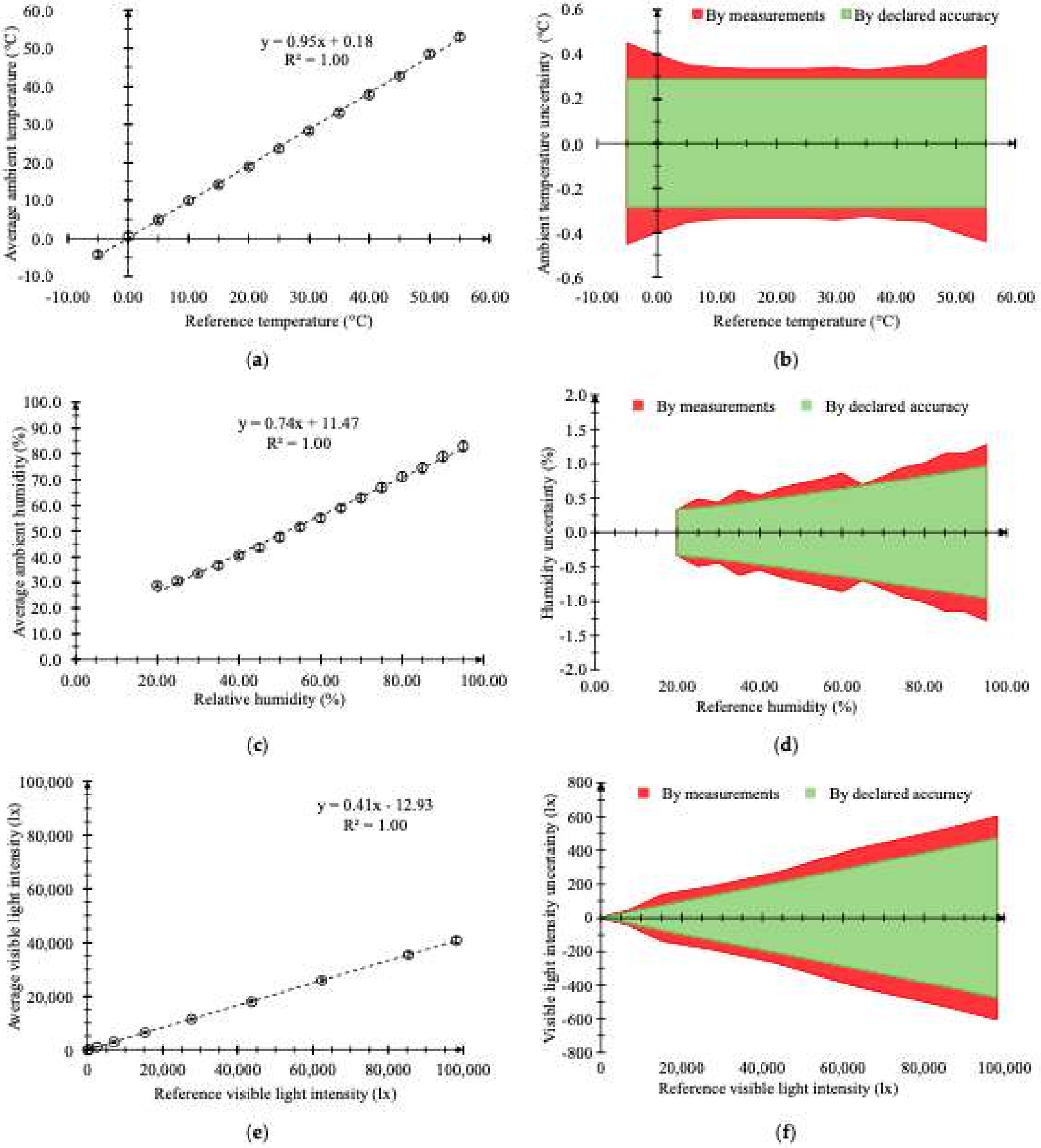


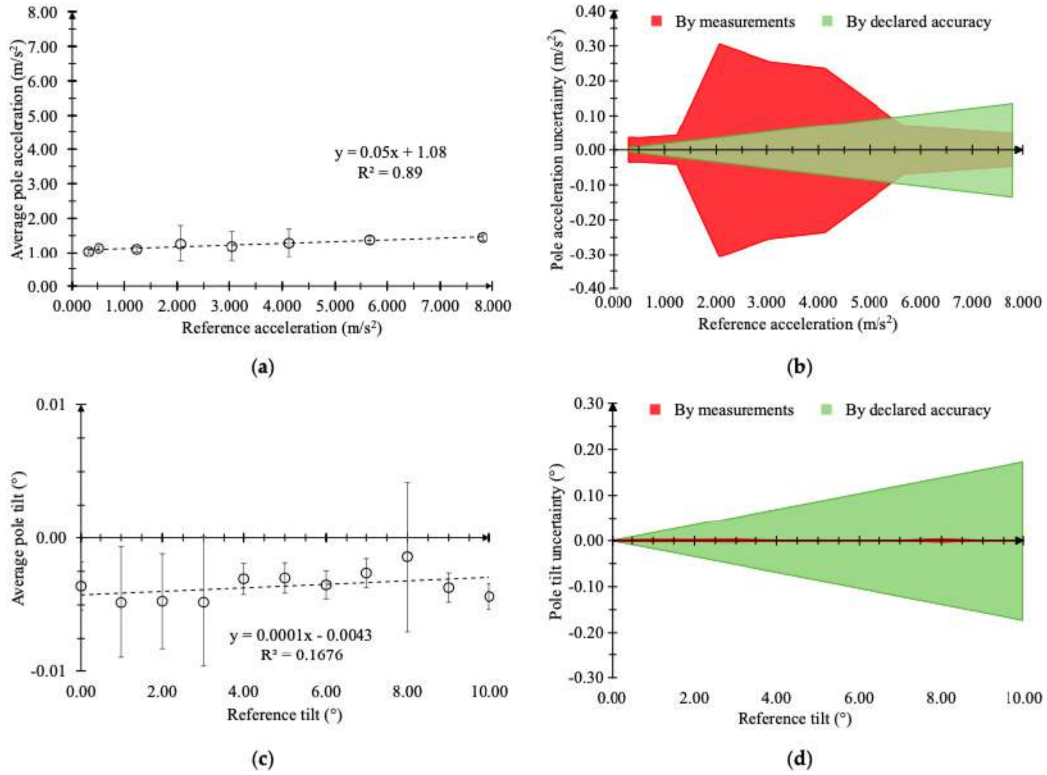
Figure 5.21: Calibration curves with bars of standard deviation and comparison between the trends of the uncertainties computed by the standard deviations of the collected measurements and by the declared accuracy for (a,b) ambient temperature at an RH of 50% and (c,d) ambient humidity at 25 ◦C, and for (e,f) visible light intensity at 50% RH and 25 ◦C of the proposed system.

The three calibration curves had a good linearity, but the collected average measurements were generally inferior to the reference ones in terms of the amplitude. Specifically, the curves of the ambient temperature and ambient humidity presented a limited divergence from the linear fitting, even if the latter of the ambient humidity was worse at the beginning of the measurement range. Instead, the sensor for the visible light intensity exhibited two different issues. On one hand, the acquired values were averagely reduced by 68% compared to the reference ones; on the other hand, the linear range of the calibration curve extended well above the maximum of 65,535 lx, as stated by the manufacturer Figure 5.13, up to 100,000 lx. Putting into evidence the trends of the uncertainty estimated by the standard deviations of the collected measurements (Figure 5.21 b,d,f), it can be seen that they were similar to those estimated by the declared accuracy. There was only an uncommon behavior in the case of the ambient temperature. In fact, the variation of the uncertainty, estimated by the standard deviations, remained sufficiently constant between 5 and 45 °C, while before and after this range, it decreased and then increased rapidly in an approximately linear way. Another observation should be made for their average values. The average values were higher than those obtained by the declared accuracy by approximately 26%, 24% and 42% for the ambient temperature, humidity and visible light intensity sensors, respectively.

For the MPU-6050 sensor in the GY-521 module, each point of the calibration curves was also obtained by considering the average value of three different measurements. However, the latter were calculated by averaging the last 10 consecutive maximum points of the acceleration signal acquired 5 s after reaching stationary conditions at a sampling frequency of 100 Hz. The uncertainty budget was achieved in the same way as reported for the previous sensors. Nevertheless, a specific consideration must be made for the vertical acceleration and tilt of the pole (Figure 5.22).

In this case, the MPU-6050 sensor in the GY-521 module displayed an abnormal behavior; although the collected signals were stable, their amplitude was rather reduced, and their variability was remarkably high. According to a detailed analysis, the calibration curves (Figure 5.22 a,c) exhibited a linear trend, but were also characterized by a fluctuating standard deviation (Figure 5.22 b,d). Specifically, for the acceleration of the pole (Figure 5.22 b), the uncertainty estimated by the standard deviations of the collected measurements was generally wider than that by the declared accuracy and above all did not have a defined trend, while for the tilt of the pole (Figure 5.22 d), the first was significantly lower than the second one. According to the explained reasons, we proceeded by replacing the employed GY-521 module with a second one (GY-521_N2) from the same production batch and then re-evaluated its calibration curves, standard deviations and uncertainty components (Figure 5.23).

The calibration curves still confirmed a linear trend, but contrary to what was previously obtained in Figure 5.22 a,c, the coefficients of the linear regression were significantly different, and the standard deviations were decisively limited and stable (Figure 5.23 a,c). Comparing the uncertainty components, their trends (Figure 5.23 b,d) appeared to be very changed from those reported in Figure 5.22 b,d. In fact, the uncertainty estimated by the standard deviations of the collected measurements of the MPU-6050 sensor in the GY-521_N2 module was only



(a)

(b)

(c)

(d)

Figure 5.22: Calibration curves with bars of standard deviation and comparison between the trends of the uncertainties computed by the standard deviations of the collected measurements and by the declared accuracy for the MPU-6050 sensor in the GY-521 module for the vertical acceleration (a,b) and tilt (c,d) of the pole at 50% RH and 25 °C.

slightly higher than the uncertainty estimated by the declared accuracy by approximately 18% and 27% for the pole acceleration and tilt sensor, respectively.

5.2.4 Final Remarks

This work presented the design, implementation, and experimental characterization of a low-cost system for monitoring the structural behavior of street lighting poles in Smart Cities. Specifically, it highlighted and investigated the benefits and drawbacks concerning economic aspects, including those of the field of information technology and the science of measurements.

Contrary to many commercial systems, the proposed one was assembled by employing low-cost equipment (DHT22, GY-302, GY-521, etc.) and open-source software (Raspian OS, InfluxDB, Grafana, etc.). Furthermore, the proposed system can be easily integrated into current infrastructures because they are characterized by a modular configuration according to the used single-board computer.

Finally, the proposed system can be employed as a starting point to ensure the achievement of metrologically detailed structural monitoring on a large scale with low-cost equipment.

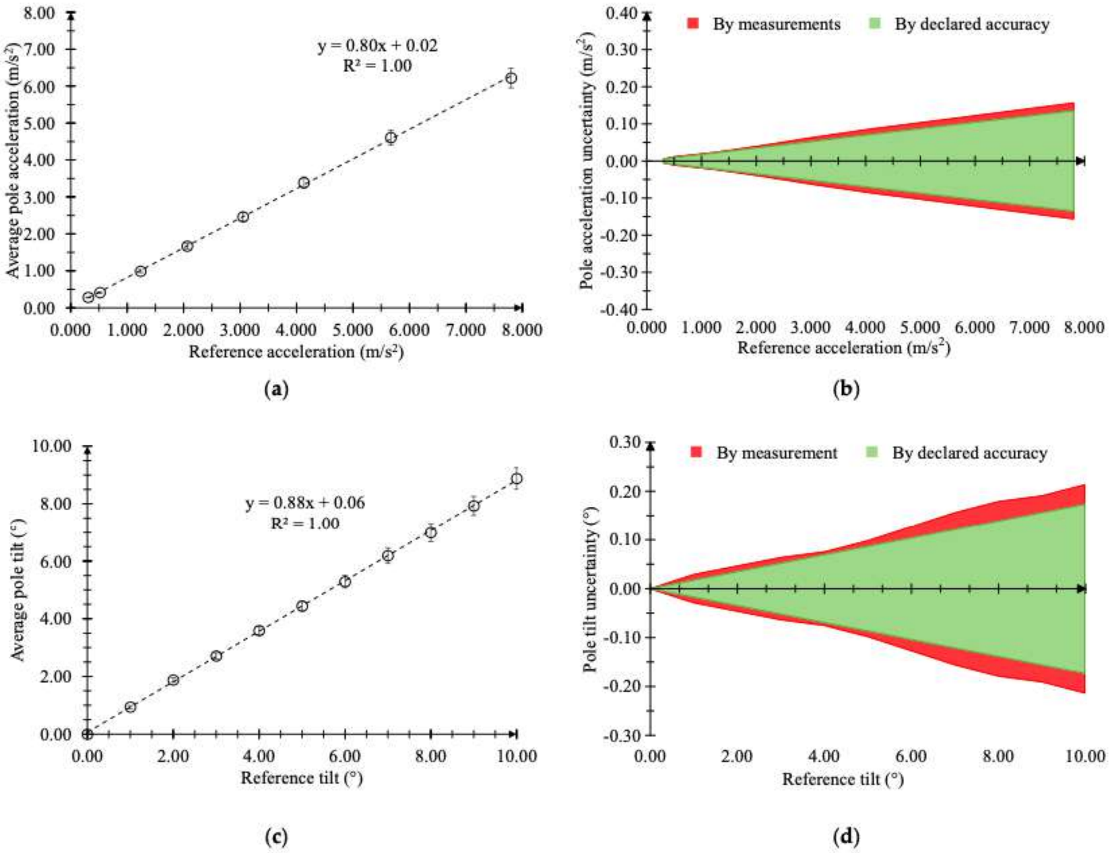


Figure 5.23: Calibration curves with bars of standard deviation and comparison between the trends of the uncertainties computed by the standard deviations of the collected measurements and by the declared accuracy for the MPU-6050 sensor in the GY-521_N2 module for the vertical acceleration (a,b) and tilt (c,d) of the pole at 50% RH and 25 °C.

Future developments will be focused on the assessment of the operational reliability of the investigated system and on the definition of a methodology for predicting the collapse of street lighting poles. Then, a large-scale monitoring campaign will be carried out on both temporal and spatial terms. Furthermore, suitable alerts for the correct prediction of the collapse of such poles, as a consequence of dangerous human activity and environmental influences, will be identified.

CHAPTER 6

Big Data Visualization in Smart Environments

The urban transformation and the changes that the world is undergoing lead, today more than ever, to the need to make faster and more timely choices in the field of mobility management. Technology is therefore essential for providing decision support tools that help managers and politicians to better manage cities. The European Commission (EC) aims to put in place sustainable mobility with the support of disruptive and innovative technologies for this sector. To do this EC financed many project, one is URBANITE (Supporting the decision-making in URBAN transformation with the use of disruptive Technologies). This chapter describes the URBANITE project with reference to the technologies and strategies implemented in the city of Messina. As a partner and pilot use case, in the municipality of Messina, software tools have been created starting from a series of local data regarding traffic, public transport tracking, and Point of Interests (PoIs) position. To accomplish the tools was important use the data of the municipality. The data for the PoIs was in static documents with specific geo-coordinate systems not useful for the rapresentation on the map. How basis for the tools an algorithm for the conversion of the geocoordinate was created and tested in different smart city applications.

6.1 Big Data Visualization Tools in Smart Mobility Scenario

In the context of Smart Cities it is crucial to pay attention to issues relating to mobility. Today Smart Mobility allows people to optimize their travels by reducing the stress associated with them, while Sustainable Mobility helps to protect the environment by improving the quality of life in Smart Cities. Institutions around the world are implementing policies that allow to decrease CO2 emissions. The issues of mobility and its optimization are therefore protagonists in the identification of these policies. In particular, the European Commission encourages projects in the field of Smart Mobility and Sustainable Mobility with H2020, Horizon Europe and the Next Generation EU programs. The URBANITE project was financed within the H2020 funding program. Among the objectives of URBANITE the main one is to promote the use of disruptive technologies in the nascent Smart Cities in technological terms through the use and analysis of Big Data, AI algorithms, etc. An innovative element, however, is that related to the promotion of innovative tools for participatory decision-making processes such as the Laboratory Social Policy (SoPoLab). The aim of the project is to provide the Stakeholders of the project with a series of innovative technological tools in order to support the decision-making processes of managers of public administrations and companies. Within the project there are four pilot cities: Amsterdam, Bilbao, Messina and Helsinki. In each of the pilots, the needs are studied and analysis tools developed which will then be applied to each of them. As regards the city of Messina, analysis were conducted on traffic and its effects on local public transport. This work describes the reference scenario and the actions implemented for the municipality of Messina within the URBANITE project regarding the purely Information Computer Technology (ICT) aspect.

6.1.1 State of the Art

In [44] a case study concerning the home-office mobility of the University of Messina staff is discussed. The home-work commuting of public employees in the city of Messina is one of the main critical issues related to daily life. Traveling at particular times of the day causes both traffic congestion and pollution. Authors analyze different performance indicators to be used for the design and development of Smart Mobility services by adopting FIWARE technologies. After analyzing the travel habits of workers at the University of Messina, authors described how FIWARE can lead to an agile development of Smart Mobility services capable of minimizing traffic congestion, fuel consumption and CO2 emissions. In [42] authors describe the results of a Sustainable Mobility project in Messina. The presented application aims to encourage citizens to use low-impact vehicles instead of private cars. Through a partnership between different stakeholders a digital application to assign citizens electric bikes was developed, free of charge for a limited time period. Authors describe cyber security issues, both in terms of secure authentication for citizens that access the service and tracking of the whole assignment process. The flow is described from the user's request to the e-bike restitution. The adopted solution uses two-factor authentication (2FA) and Blockchain as the main technologies in the implementation phase. Innovative and advanced smart devices and virtual devices are described in [43]. Authors have designed, for one use case in the city of Messina, an abstracted component characterized by specific high-level functionalities. The system offers the chance to access the needed information with the most appropriate frequency and accuracy, avoiding information overload and allowing a more efficient computation. In this case it is important the access control and the security of the data. An interesting work for this purpose is described in [2]. In [148] authors show the use of customized generic Edge devices to carry out multiple activities at the same time, also focusing on how the proposed solution can lighten the work of cloud infrastructures. The presented concepts were implemented and tested in a real use case in the city of Messina by means Function as a Service (FaaS) paradigm. The proposed work allows users to perform multiple tasks on the same device. Applications such as vehicle counting, license plate recognition, object identification, etc. are proposed. In the considered use case two cameras were connected to a Raspberry PI 4 and the performance was compared. It is possible to connect different sensors to the proposed Edge devices and imagine each sensor as a different service. In [149] authors introduce a tool for studying mobility data. The basic principle is that technological innovation has led to the spread of various data tracking systems. The data are accumulated and can be used in various applications such as the analysis of mobility, urban planning and transport engineering. It is possible to use the data to extract information in matters relating to rough space-time trajectories, or by relying on statistical "laws" governing human movements [150]. However, authors do not neglect the attention to user privacy [151]. From the study and development comes an interesting Python library used in URBANITE for the analysis of mobility data in particular in Messina use case. From the state of the art it emerges that the city of Messina has been the subject of various scientific studies that have found practical application. Various national and European grants made it possible to achieve relevant innovations in the field of mobility. It is not clear how the data collected can be useful to administrators and managers in the decision making phase. This work, therefore, want to synthesize and demonstrate how, thanks to URBANITE project, it is possible to put together what is already present in the systems of the city of Messina, creating the basis for the creation of new useful decision-making tools.

6.1.2 Reference Scenario

The URBANITE project was created to provide communities with a long-term sustainable ecosystem model. Through a co-creation strategy we want to bring stakeholders (civil servants, citizens, etc.) closer to the use of disruptive technologies in the field of mobility. This model is supported with a data management platform and algorithms for data-driven decision making in the field of urban transformation. Furthermore, the model is validated by pilot mobility use cases in the context of the proliferation of sharing services. The URBANITE platform encapsulates the experiences of four pilot cities and acts as a junction point to create a unique analysis model for cities. Thanks to the platform it will be possible to have information regarding mobility that can be as a support in order to take serious technical and practical decisions.

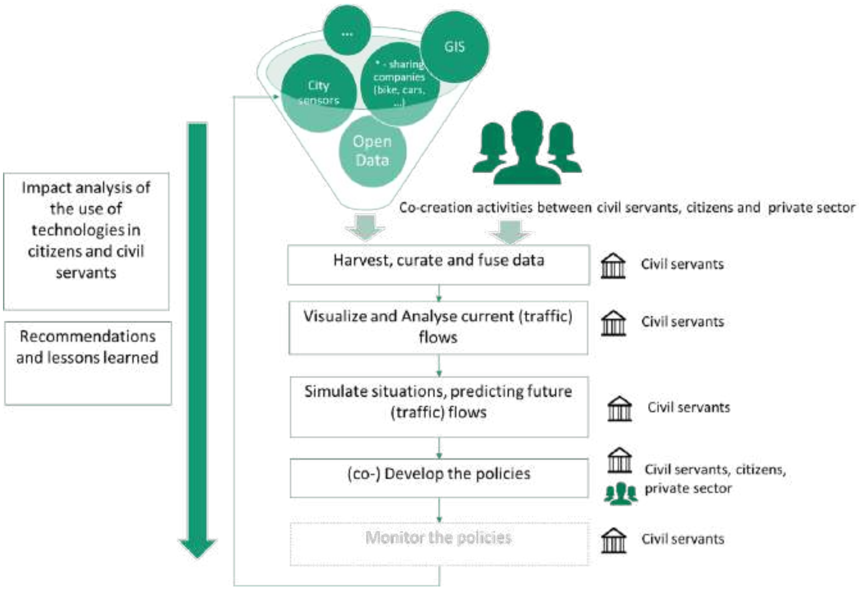

...
GIS
City sensors
* - sharing companies (bike, cars, ...)
Open Data
Impact analysis of the use of technologies in citizens and civil servants
Recommendations and lessons learned
Co-creation activities between civil servants, citizens and private sector
Harvest, curate and fuse data
Civil servants
Visualize and Analyse current (traffic) flows
Civil servants
Simulate situations, predicting future (traffic) flows
Civil servants
(co-) Develop the policies
Civil servants, citizens, private sector
Monitor the policies
Civil servants

Figure 6.1: Urbanite Approach.

In each pilot the data, useful for the mobility analysis, were analyzed and collected. The data considered funcional are collected on a single data storage. Thanks to different visualization and AI techniques/algorithms, the data were processed and made possible to create decision making tools that currently need validation (Figure 6.1). The use case regarding the city of Messina is described below.

Briefly on Messina Use Case

The metropolitan area of Messina is one of the most extended areas of the south of Italy, the first in Sicily and counts over 620.000 citizens. The city counts over 250.000 citizens and most of them are commuters between Sicily and Calabria. The local transport of the city of Messina consists of both sea transport (hydrofoil and ferry boats fleets) and land transport (buses, tramway and rail transports network). They are managed by public and private companies. The main issue that affects both kinds of services (sea and land transport) is the lack of facilities that can permit interoperability between different departments of the municipality and the communication with citizens and stakeholders. In order to overcome this problem, the Municipality of Messina is investing in intelligent infrastructures and services for the city and citizens. In particular, the main activities are focused on vehicle access detection in LTZ (Limited Traffic Zone) and pedestrian areas, centralised traffic management based on smart lights, traffic flows and analysis, incentives to use public transportation and video surveillance. URBANITE, for the city of Messina, is focusing on light and pedestrian mobility. Concerning the light mobility there are two main action lines:

1. the extension of the cycle paths and the spread of bike mobility (but the main goal is to promote the use of bicycles and to offer better services to citizens)
2. create new bike-lines and links between the centre and suburbs zones of the city.

Regarding pedestrian mobility, the objective is the definition of an integrated system of pedestrian areas and paths. Furthermore, from a wider perspective concerning public transportation, the city of Messina aims to extend the transport network in urban and extra-urban areas. The use case scenario in Messina (Figure 10.6) aims to evaluate the effects of the extensions of the public transportation services in terms of frequency, itineraries and stops on traffic and multi-modal transportation. In particular, a comparison of the impact on traffic between the different version of the public transportation network was performed. Moreover, the scenario includes an analysis of the suburban roads around the city of Messina (that represent an important connection with the surrounding towns) in terms of traffic congestion and connection with public transport network.

The URBANITE Architecture

The architecture created within URBANITE is made up of several abstract components that interact with each other. Thanks to the interaction between the different components, it is possible to provide all the tools necessary to achieve the objectives of the project. In Messina this architecture has been enriched by building new dedicated components, at the Edge level, which fully integrate with the existing Cloud ecosystem as shown in Figure 6.3, in which these components are highlighted.

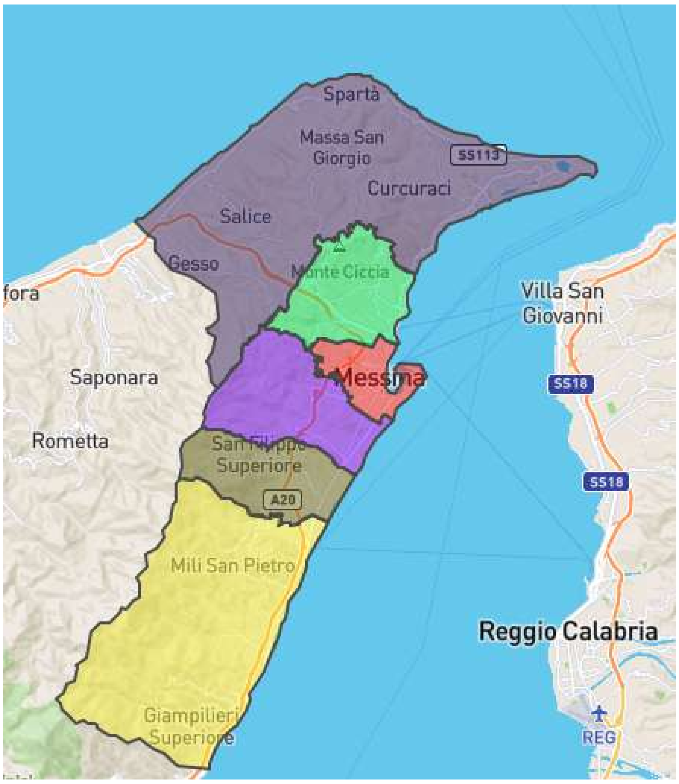


Figure 6.2: City of Messina District.

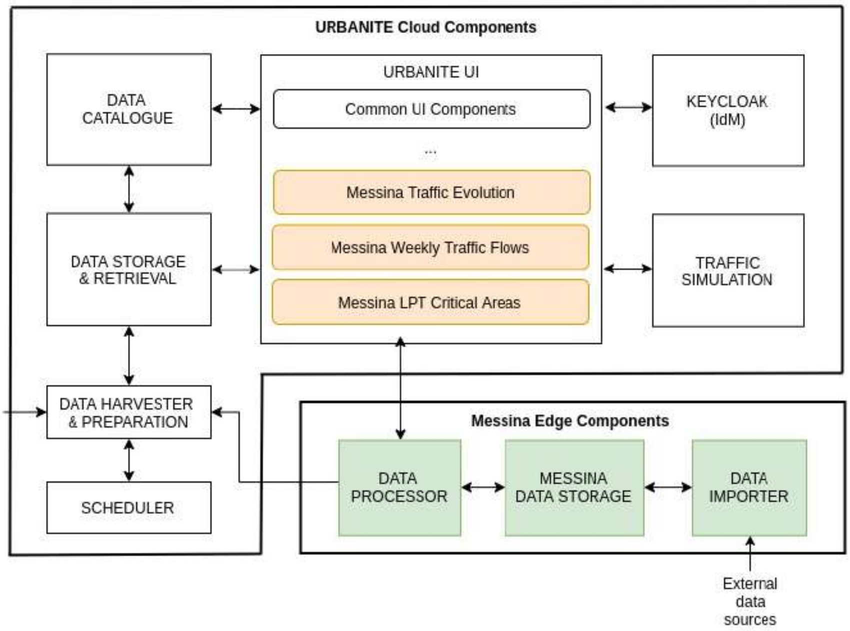
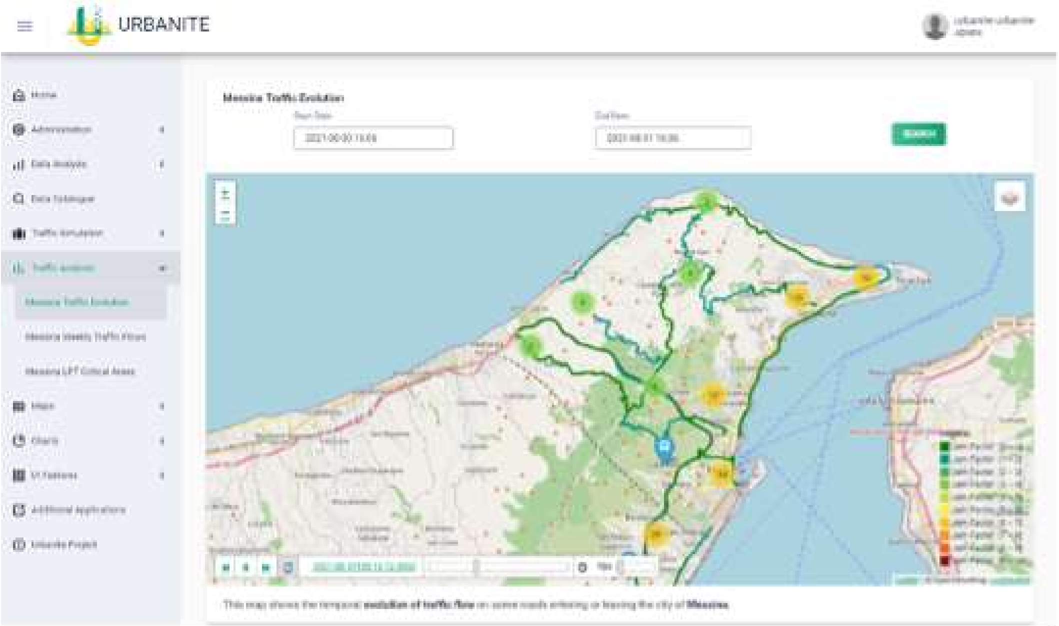


Figure 6.3: URBANITE Architecture - Messina.

In particular, for the Messina Edge Components, a local component called *Messina Data Storage* has been added. This component acts as a support for the parent component *Data Storage & Retrieval* (reported in URBANITE Cloud Components) through the *Data Harvester & Preparation* and is filled with data by the *Data Importer*. The *Data Processor* allows both to expose the data via Restful API and to process them ensuring correct formatting. Finally, within the *Urbanite UI*, three new specific components for the Messina use case have been built: *Messina Traffic Evolution*, *Messina Traffic Flows*, *Messina LPT Critical Areas*.

6.1.3 Messina Implementation

The use case scenarios described before are accessible thanks to the functionalities provided by the URBANITE UI, the integrated URBANITE's framework at the UI level. The different analysis and visualizations provided aim to help the municipality's technicians in the extension of the current public transportation network. The tools allowing the users to interact with each visualization by filtering and querying the underlying data. Concerning the traffic congestion analysis for the municipality of Messina, Figure 6.4 depicts the temporal evolution of traffic flow on selected roads entering or leaving the city of Messina.



This map shows the temporal **evolution of traffic flow** on some roads entering or leaving the city of **Messina**.

Figure 6.4: Messina Traffic Evolution.

The traffic jam factor of each road, in a specific time of the day, is represented by the colour of the road itself, following the provided legend. Data used to this purpose are acquired and stored for real-time and historic analysis. Figure 6.5 illustrates the comparison analysis of the jam factors on two different roads of the city considering the time window of a week.

The data source is the same of the previous analysis, but this time the purpose and the target users are people with a more technical background. For each road, if the road is bidirectional, the dashboard provides a chart for each direction using a different symbol for each one. The colors indicate the jam factor value. Finally, to identify areas of Messina where vehicles of public transportation are stationary for a certain time in a specific observation

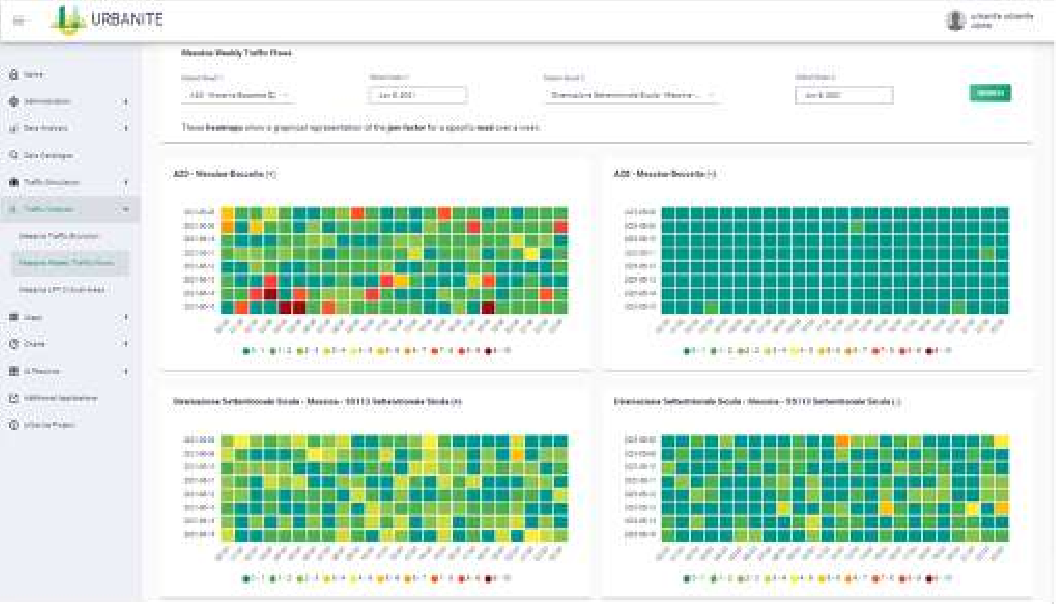
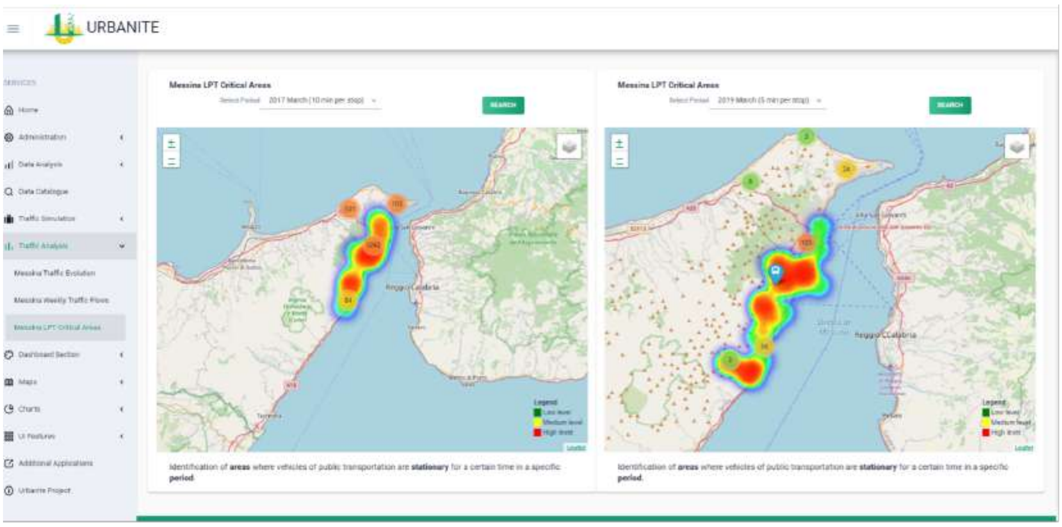

URBANITE
Messina Weekly Traffic Flows
A20 - Messina-Boccetta (+)
A20 - Messina-Boccetta (-)
SEARCH

Figure 6.5: Messina Weekly Traffic Flows.

period, the heat-map analysis, depicted in Figure 6.6, is provided.



Messina LPT Critical Areas

Select Period: 2017 March (10 min per stop)

SEARCH

Identification of **areas** where vehicles of public transportation are **stationary** for a certain time in a specific **period**.

Messina LPT Critical Areas

Select Period: 2019 March (5 min per stop)

SEARCH

Identification of **areas** where vehicles of public transportation are **stationary** for a certain time in a specific **period**.

Figure 6.6: Messina LPT Critical Areas.

To investigate if public transportation means use to be stationary in the same place for different time periods, the dashboard allows to compare two different time slots. In this case the data source is an historic database for the bus and tram position of the Local Transport Company. The data are elaborated with the scikit-mobility [149] Python library with the aim to obtain the heat-map visualization. In each described visualization, in order to have further information, the dashboard allows to visualize Points of Interest and Public Transport Stops on the map.

6.1.4 Final Remarks

This section describes the current state of the ICT systems put in place for the URBANITE project as regards the case of the Messina pilot. From the first results it is evident that, thanks to the use of data analysis and their appropriate visualization, it is possible to obtain information that is often difficult to understand. The visualization methods allow for immediate analysis and support decision-making policies. Thanks to the presented tools, in fact, it is possible to determine the effectiveness of the mobility policies used compared to the past, thanks to the historical harvested data, and possibly try to improve them. The next step will be to extend the functionalities. The scenario of each single pilot must be applied to all the case studies of the project. Moreover, it is necessary to improve smart algorithms in order to have responsive systems even in real-time. Finally, the system will make the APIs available for open-data, giving other scholars or stakeholders the possibility to carry out analysis or develop innovative solutions.

6.2 Smart City Use Case for geo-coordinate conversion

Smart Cities offer different services to citizens, starting from access to offices to services management. This work aims to study the problem of WDN's digitization in a Smart City, in terms of representation on a digital map like Google Maps. The digital representation of a water network can be made with different real-time visualizing flow tools. Nowadays, the use of 2D or 3D GIS systems is largely rising in various projects for managing and safeguarding the city's water resources. For example, FIWARE4Water [152] intends to connect software used in the water sector to the FIWARE ecosystem. In Europe, digital-water.city [153] aims to promote the integrated management of water systems in five major European cities: Berlin, Copenhagen, Milan, Paris, and Sofia. The project, driven by the potential of data and AI, aims to develop innovative solutions in close collaboration with Public Administrations (PA). These projects try to apply web or GIS-based solutions to solve the problems faced in a WDN and include funds for specific research areas. Despite the different solutions proposed in the literature, there aren't many tools that perform an optimized and real-time analysis of the WDN's status. The GIS-based applications with sensors give a static view and measure of WDN's flows without the possibility to georeference and monitoring each point in the network in real-time. The current challenge is to use technologies that can integrate the WDN's managing services with digital interactive visualizations tools. Moreover, using the referencing technologies on Google and the related maps allows us to easily integrate the work of technicians specialized in the design of water networks with the services for citizens possible thanks to modern ICT technologies. In particular, the integration of Google Maps tools for the WDN's representation makes possible the implementation of color visualization techniques of the pipes and displays real-time or historical graphs, simplifying the identification and reporting of leaks.

6.2.1 State of the Art

Recently, the digitization of water distribution networks has been surveyed in several scientific works. The digital representation of water networks can be done using CAD [154] and GIS [155] technologies, which are also the software most used by water engineers. In particular, it is possible to make the WDN management and visualization systems more interactive by using solutions based on the integration of ArcGIS [156] with services offered by Google. Software such as Epanet is used to design the WDN. With Epanet [157] it is possible to apply closed-loop self-correcting mathematical models, for the analysis and simulation of the hydraulic parameters of the network. Moreover, ArcGIS technology is adopted for the integration of WDN visualization with network monitoring, management, and control functions. This solution was applied inside a small Campus [158], but algorithmic and system performance analysis and interactive WDN display are not performed. One of the main problems of managing the water network inside a Smart City is the detection of leaks. To this end the Support Vector Machine (SVM) [159] and Radial Basis Function (RBF) [160] are applied inside prediction models in order to report the leaks with their position and find a solution to the water supply problem by monitoring the pressure and quality of the water.

Furthermore, to optimize water resources in a smart city, Network Flow [161] algorithms are applied within the digital WDN. These algorithms are used to monitor the flow rate, and for a preliminary analysis for the positioning of certified sensors on the WDN [162], [163]. One of the main issues affecting WDNs is the aging of [164] pipes. A possible solution consists of the matrix representation of the network that is the basis for the design of systems for water monitoring and decision support in smart cities [165] and [166]. Differently from the aforementioned scientific initiatives, which are mainly based on the use of design software like GIS and CAD, in this paper, we focus on how the Google Map technology (mixing with google functionalities, optimized algorithms, and conversion system) can be used to carry out the interactive visualization of a Water Distribution Network in a smart city. The aim of the proposed work is to present an innovative cloud based ICT system that is easily replicable and scalable. The system was born on the one hand for the development of decision support systems for the management of the companies that manage the WDNs, on the other hand to provide an ecosystem that allows to provide digital services to citizens in the context of WDNs.

6.2.2 Motivation

Nowadays, the provision of digital services for monitoring and supplying water networks through smart applications is becoming a major challenge to be faced. It is estimated that by 2025, two-thirds of the world's population could be facing a water shortage [167]. To address these issues, water consumption and digital tools for decision support and service offer need to be optimized, for example reporting leaks.

The digitized WDN becomes a support service for technicians and organizations that control a water network. In particular, the timely reporting of losses in a WDN by a citizen can facilitate the intervention of specialized technicians in the place of interest. In literature, there are several systems based on GIS, but none of these uses technologies to provide a real-time "photograph" of the network's state.

The Google Map technology is used for a digital representation of a WDN and can allow the technician to have a real-time view of the WDN's state. In addition, by integrating the features supplied by Google, it will be possible to provide technicians with a decision support tool for maintenance operations. The design and dimensioning of a water distribution network in a smart city is a very complex operation. Various CAD software, based on GIS, are used to design and structure a network. These solutions provide a low-quality user experience and scalability. However, most cities, and/or organizations that manage water services, do not have digital documentation, or CAD projects that represent the distribution network. In addition, the network is subject to structural changes, maintenance, and replacements that are not updated in the network design files. These updates can drastically affect the information that is provided in inputs to systems of representation, acquisition, and management of water distribution networks. Using a Google representation integrated with sensors that ensure reliable data transmission [2] can be a solution to these problems and, can allow future network design directly on the digitized WDN. From a topological point of view, the WDN can be associated with a graph, where each node represents a junction, a tank or a valve, and the arcs are the pipes between the nodes. With our system, we want to propose a framework for the interactive visualization of WDNs on Google Map. To achieve this goal it is necessary to implement conversion algorithms that allow converting the coordinate systems of the design software with a WGS84 representation. Algorithms are run and data are processed in the Cloud. By integrating the innovative features provided by Google with our analysis and conversion algorithms, it will be possible to give users a detailed view of a WDN on a Google map. In literature, there are different formulas and mathematical methods that allow us to carry out conversions, but none of those tested allowed us to apply an effective conversion model for the proposed system. For the conversion and processing of input data to the conversion algorithms, it was necessary to structure the data present in the design source file accordingly. In this way, starting from a file extracted from Epanet, it is possible to get an interactive representation of the network on a Google Map. In addition, it will be possible to integrate data visualization algorithms that will allow representing the state of the networks in real-time. In this work, by integrating the interactive Google map with newly defined criteria for data visualization, it will be possible to associate a color to a critical state within the network and visualize the data acquired in real-time by the sensor installed on the WDN nodes. The tool that will be described will guarantee services to both technicians and citizens, and the scalability of the system in terms of load data management will be demonstrated.

6.2.3 Design

The proposed work aims to demonstrate how the cloud can support the provision of innovative services in the field of WDN management. In particular, the proposed system has the purpose of optimizing the digitization of a project created with a CAD software, creating an optimal representation to be stored in a No-SQL database and which can then be exploited for a cloud-based management of the system. Our solution implements an infrastructure that integrating Epanet Software, Google Maps service, and Cloud systems, as shown in Figure 6.7.

EPANET is the software used for modeling water distribution systems. Through Epanet it is possible to design the WDN with different distribution elements like junctions, valves, tanks, and pipes. From Epanet also it is possible to simulate and extract models in order to integrate them into sophisticated processing systems.

Cloud all the conversion and visualization algorithms of the WDNs on a Google map are performed in the Cloud. The proposed system is designed in a way that in the cloud it is possible to manage backend services, such as interfacing, user accounting, conversion

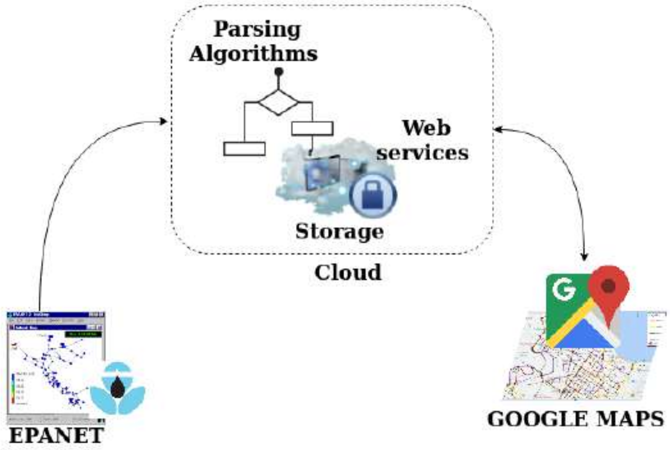


Figure 6.7: System abstraction.

systems, and processing of complex data structures. Furthermore, in the Cloud, there are web servers and storage systems for real-time monitoring of the data acquired by the sensors in the WDN.

Google Maps for a complete view of the WDN, through the API of Google Maps, it is possible to enrich the interactive maps with additional services. The mapping of the services provided in a water distribution network can be done by using specific markers. To view the pipes of a WDN and its status, monitored by sensors, APIs are used to provide the user with information about the state of the network in real-time.

To convert the data extracted from the Epanet model into Google Maps WGS84 coordinates, it is necessary to carry out the conversion that characterizes the entire system proposed in this work. A .inp file executable with particular input software is extracted from Epanet and applied to carry out simulations on the network. Through the application of implemented conversion algorithms, it was possible to report the position of a generic junction in the Google map. The conversion algorithms are based on complex data structures characterized by abstraction where the WDN is a graph where each node is a junction, and the arcs are pipes with a weight.

The proposed system, Figure 10.1 is based on a multi-level microservices architecture: Edge Layer, Cloud Layer, Frontend and Backend Layer.

- The Edge Layer provides the user with the graphical interface of the WDN and communicates with the other layers using the REST APIs;
- The Cloud Layer contains both the WebAPP and the Backend system. In the Backend, there are two main blocks: Identity Access Management and Computation Services;

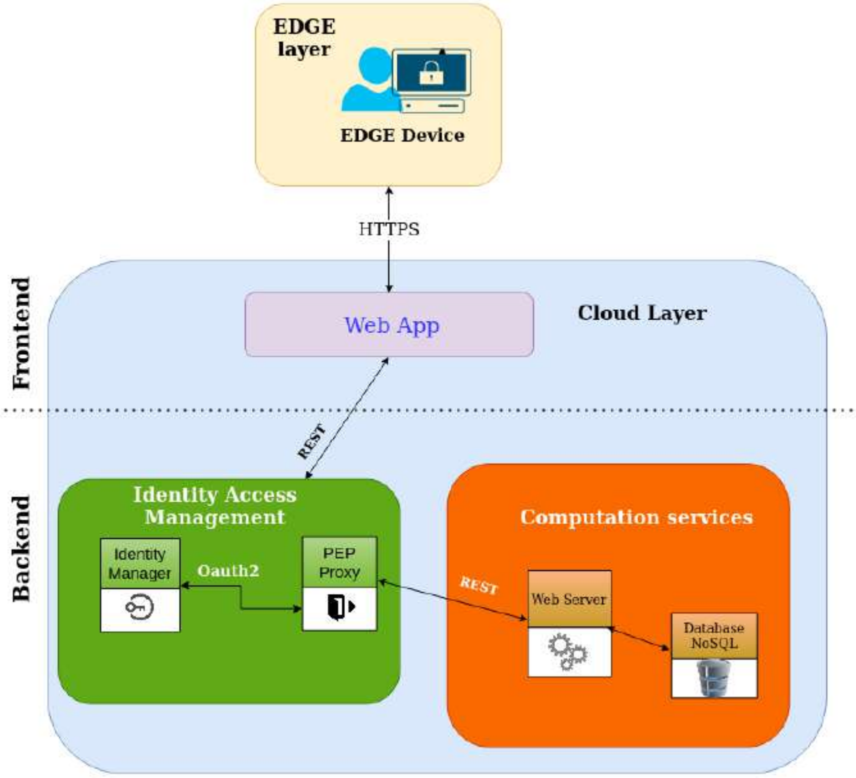


Figure 6.8: System Architecture.

All requests, authorizations, and user accesses are managed in the Identity Access Management. Only authorized users who have received a valid Oauth2 token can access the system. The Computation Services block allows us to perform all the analyzes and coordinate the conversion algorithms, storing all the data in the appropriate NoSQL databases. All interactions between the user and the system take place using authorized requests. The data is not exposed directly as the HTTPS and REST protocols have been used.

6.2.4 Implementation

In this section we present the conversion algorithm developed for a smart representation of WDN networks, with particular attention to the data structure underlying the WGS84 coordinates. Furthermore, the functionalities implemented to offer highly innovative cloud services will be presented.

File Parsing & Junction Coordinate Conversion

To extract the data of a WDN from an EPANET file and represent them on a Google map, it was necessary to use the EPANETTOLS library. Using this library, a parser for the EPANET ".inp" file was created and the algorithms for real-time management of flows were designed. Furthermore, the first algorithmic step consists in converting the spatial coordinates extracted from the EPANET file into WGS84 coordinates (latitude and longitude). To do this it was necessary to build a specific adjacency list in which each element is a tuple (id _node, X-Coord, Y-COORD). Once the network has been represented, using the WGS84 standard it is possible to calculate the coordinates of a point starting from a known point and consider the sphericity of the earth's surface. This is possible with the Python geopy library, which provides the "VincentyDistance" function. As previously described in the design section, the conversion process starts from the known node and then the coordinates of the first neighbor are calculated and repeated on all the nodes of the network. Nodes already converted are not considered in the next iteration. The iterations end when there are no more nodes to convert and the network has been fully explored.

Database creation

In addition to having a complete representation of the nodes in the network, it is also necessary to record the information on the pipes in the database. To do this, it is necessary to know the coordinates in WGS84 of the nodes at the ends of a pipe, the maximum flow rate, and the measurements acquired by the sensors in terms of capacity (actualCap) and residual capacity.

Listing 6.1: Creation of a collection containing all the pipes of a water network

```
import pymongo
# Connection with MongoClient
from pymongo import MongoClient
client = MongoClient('mongodb://172.17.0.2:27017/')
db = client.amam  #use db
#inset Pipe in db
def insertPipe():
    #idPipe = 0
    f=open("graph.txt","r")
    for line in f: #read line in file
        raw = line.split()
        #add documents in the collection with a dictionary
        mydict = {"idPipe": raw[0], "sourceNode": raw[1],"destNode": raw[2], "
    maxCap": raw[3],"actualCap": 0, "residualCap":0}
        insert=db.pipes.insert_one(mydict) #insert the elements
    f.close()
insertPipe()
```

To create a specific collection in the database, we used the script shown in Listing 6.1, which reads from the EPANET text file containing the WDN's list of edges. For each line of the file, it writes the information to the database as previously described. Moreover, to have a detailed view of the network we carry out some simulations on consumption and groups of users.

From a technical point of view, the users are part of the graph representing the water network. We chose to define three types of water supply group of users identified as "Residential", "Commercial", and "Industrial". Considering that the total number of consumers for the city of Messina is reasonably around 120,000, each group will consist of a maximum of 50 users, 60% of which are residential, 30% commercial, and 10% industrial. To implement the algorithm for user creation is necessary to store all the information and converted nodes in WGS84 coordinates on the database. For the prototyped system proposed in this work, we chose to generate randomly the number of consumers per supply group, insert the source junction for each leaf, and finally evaluate the total flow into the leaf. The sum of the flow rates for the pipes entering the junction corresponds to the water flow with which the group of users connected to the junction is fed.

Synthetic data generation: Real time simulation and data history

We have implemented different functions to simulate the flow control systems in the pipes and make them visible at the frontend level. The proposed system needs the installation of sensors that measure the water flow level in real-time, but from a technical point of view, we choose to simulate this process to have a history with one-year measurements of water flow. For each measurement of each pipe, we evaluate randomly the flow value and a timestamp that is associated with them. To simulate the behavior of the network and Big Data we have created a history of data flows for a period of one year. As reported in [161] it is recommended to perform the flow measurement every 15 minutes. In this way, it is necessary to simulate 96 daily readings for each tube for 365 days. With a WDN of 4447 pipes, you need to create collections with 155 million documents. For each measurement of each pipe, we randomly calculate the flow rate value and an associated timestamp.

Frontend development

For the visualization of the WDN on Google Maps, we have implemented a frontend application to safeguard all the sensitive data of a water network in a smart city. In particular, an RBAC system was implemented with FIWARE IAM for the management of roles and authorization of users and technicians. In Figure 6.9 is reported a representation of a city water network.

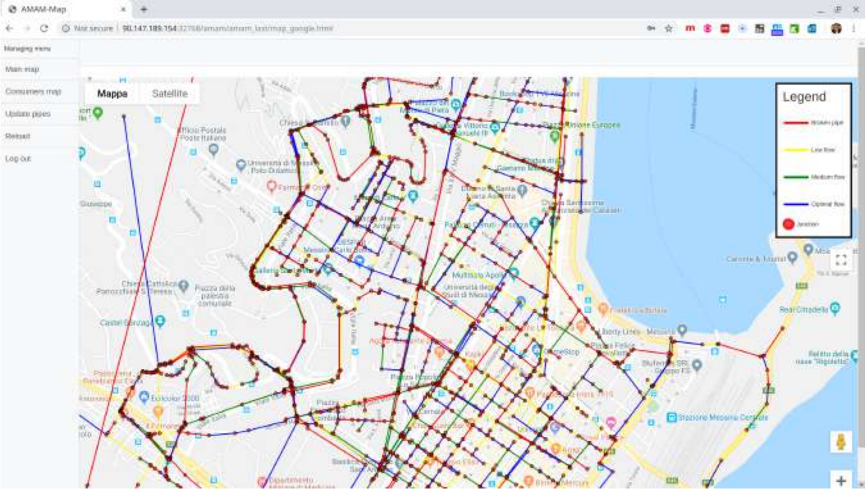


Figure 6.9: Messina WDN rapresentation.

Through the Google libraries it was possible to implement different functions on the network. Furthermore, to display the utilities inside the WDN it was necessary to use a bit cluster connected to the junctions (Figure 6.10).

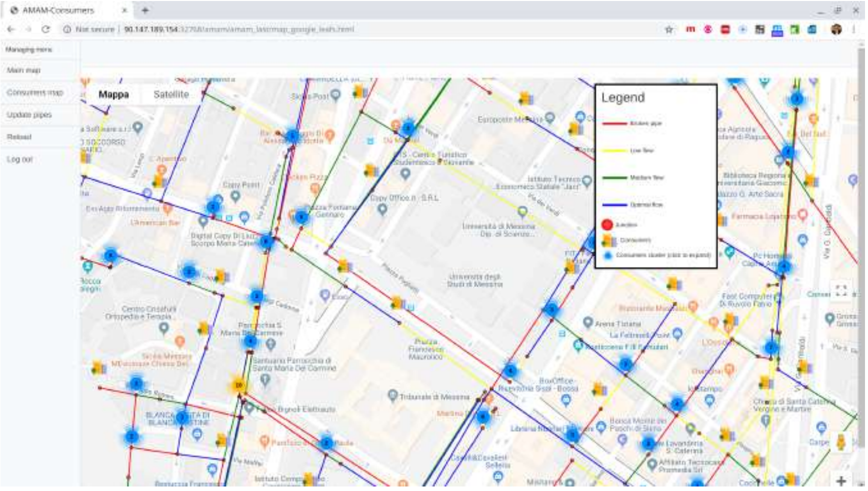


Figure 6.10: Messina consumer map.

The data necessary for the reconstruction of the maps are extracted from the backend through AJAX calls. Calls are made to a public address, but all are authenticated through the FIWARE GEs Wilma token. This token is issued during login and is then kept for use in different calls. Using these procedures allows us to make the system more secure. By selecting the user group, it is possible to view the trend of the water flow for the users in the 8 readings preceding the one selected, corresponding to the previous 2 hours, (Fig. 6.11).

In the different simulations carried out, the percentage of water flow directly on the front

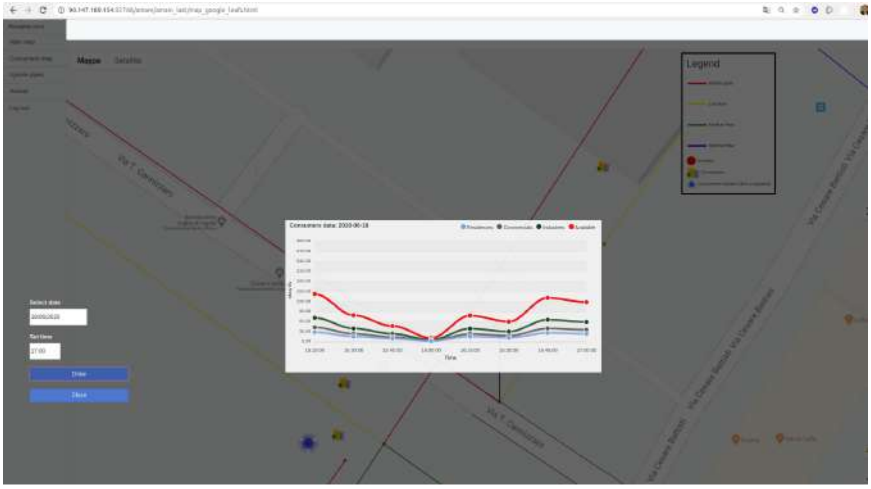


Figure 6.11: Example of flow over time for a group of consumers.

end was evaluated in order to show consumption by type of user in a given time band.

The graphics were created using the Apexcharts JavaScript library, while the display base is configured via CSS. Bypassing the mouse over the points on the graph, the reading will be automatically enlarged giving more details to the user.

6.2.5 Experiments

In this section we report the system scalability tests. To test the system's ability to work under stress we use JMeter software. JMeter allows us to send a large number of requests simultaneously to the webserver. The maximum number of requests is fixed at 500. This number comes from the limit of maximum requests allowed by the proxy to protect the system from a single ip address in a limited time interval. Calls to three endpoints were tested: junction request, request pipe and water utility request. Furthermore, in this section we report some considerations on the various data management procedures obtained during the development phase

Test Bed Setup

In our experiments, we implement the features described in the previous sections using two virtual machines hosted on the Garr network with nodes in Palermo (Italy). The backend of the system is located on a Virtual Machine with 4 GB RAM, 40 GB HD and 2 VCPU Intel Xeon 64 bits, 2 GHz.

Results

In Fig. 6.12 we report the increase in memory request on the machine in the interval corresponding to the three request flows.

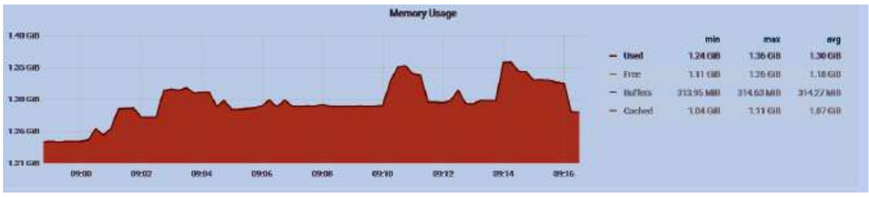


Figure 6.12: Memory usage request in case of flow of 500 get requests.

The use of memory is not affected by any problems as expected. No changes were reported on the use of the processor. For a more in-depth analysis, we monitored the use of resources by the container in which the PEP Proxy Wilma is running.

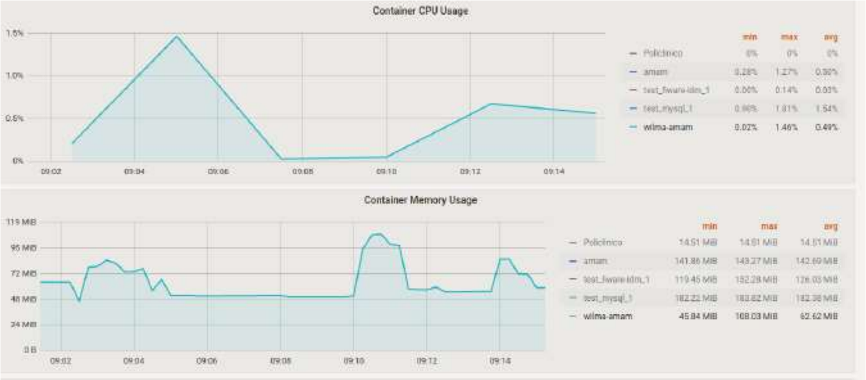


Figure 6.13: Resource monitoring for the Wilma Pep Proxy container in case of flow of 500 get requests.

In Figure 6.13 we report the use of memory and processor by the container. As already noted above, there are no major variations in the use of resources. The minimal and not significant variations in the use of resources allows us to say that the system, as expected, has a good ability to work under stress. There are no graphs on the use of database resources or other processes because, to confirm the correct functioning of the system, there are no significant changes.These considerations also indicate the correctness of the optimization procedures put in place.

Data manipulation

Tests of the affected system and creation of a data history are performed. In addition to the fact that the data are simulated, even in the presence of real data acquired through sensors, it would be necessary to create algorithms to insert them into the storage structures. The implementations of these algorithms influence the data writing times and therefore a possible system recovery in case of failure. To understand how writing times are affected, in the case of big data, we tested two different approaches. The estimated result of running the two scripts, which had to write about 155 million documents in a mongoDB collection, was that the first case takes about 30 days to complete writing to the database, while the second case takes about 2 hours and 30 minutes. The differences are due to the fact that, in the first case, at each iteration, a read and then write access is made to the database. The second case considerably reduces the execution times, since all the reads are performed before the iterations, the operations during the iterations are performed directly in RAM, and every 4447 iterations a write is made in the database that corresponds to a measurement on all the pipes.

Querying consideration

Another test was performed on the database concerning requests for the extraction of data to be sent to the front end. In this phase, we verified that without the use of MongoDB indexing, a query on a database of about 155 million documents takes such a long time to send the requesting client in timeout. An indexed MongoDB database optimization uses time in the order of a few seconds for queries. Another performance comment is that a query should attempt to filter out as much data as possible to send to the front end. If the data is not filtered directly by the query, it can happen that sending large amounts of data sends the client to time out. Through testing, we have found that if a large database is not indexed, queries take a long time to run. On the other hand, creating indexes on a large database like the one under consideration takes a long time, in our case about 1 hour and 30 minutes. We consider this time tolerable since without the indexes it would not have been possible to create a client-side application. Furthermore, the creation of the indices is a backend operation that, given the case study, must be carried out once or in any case only when there are large variations in the water network.

Data Visualization

Through testing on our system, we realized that the source of a large amount of data and its processing on the client caused latency problems. This situation did not allow the user to view the desired data. In the optimization phase, we have modified the queries. In particular, by inserting the WHERE conditions among the search parameters, sorting the results, and placing a limit on the data found, we were able to obtain an optimization of the display times that significantly improved the user experience in the system proposed. A different representation of the data on the backend side allowed to reduce the time needed to analyze and view the data on the frontend side, allowing to create a prototype of the desired system.

6.2.6 Final Remarks

The section presents a solution for the digitization of water networks in smart cities. The representation obtained allows for representing the network on a digital map. The sensor data was obtained through synthetic data generation. After studying the need for a representation on Google Maps in the city of Messina, we investigated the current state of the art for the representation of WDN on digital maps that allow real-time analysis of the state of the network. In literature, there are many limitations due to the ArcGIS visualization and the technology applied in the representation of the water network. These limitations mainly concern the scalability of large amounts of data as well as the non-existent possibility of offering services to citizens. To overcome these limits we have chosen to move our studies to the representation on Google Maps of the projects carried out in the Epanet software. With this solution, it is possible to enrich the simulated network with the information detected by the sensors positioned on the WDN. Through a complete representation of the WDN, it is possible to have a visual representation of the colored pipes. In our solution, we integrate the possibility of representing groups of users and monitoring consumption by type of water user. We have described the design as the basis of our solution. Finally, we tested the performance of the system and reported some considerations made during the testing and development phases. In conclusion, for future development, we plan to build a field test environment on the system and insert sensor control mechanisms. Furthermore, we want to integrate the system with a progressive web app that can give the user the possibility to report leaks in the WDN. In this way, the intervention of the technicians will be more timely in order to significantly reduce water losses.

CHAPTER 7

Smart Technologies for Users Movement Support

The organization of travel in modern urban environments does not only concern the organization of urban public transport data and services. It is necessary that the different "entities" that operate within smart urban environments can facilitate and optimize the times necessary for the various activities. This chapter reports some use cases and applications of innovative technologies in order to facilitate the activities of users who move into an urban environment and take advantage of its physical services. Consumers in physical stores increasingly expect a personalized shopping experience and are looking for information and assistance when researching products. The retail sector has already undergone a shift towards more ubiquitous in-store technology, but intelligent shopping assistance for consumers is absent. Whether searching, locating, and navigating physical products can work with contemporary technology is an open question. We tried to give an answer to this problem by analyzing the feasibility of introducing an intelligent purchasing workflow based on product research and internal navigation in a real environment. The analysis of the problem also addressed the impact that social networks and virtual reality can have on external and internal travel if associated with technologies such as geotagging and BIM (Building Information Modeling). Another aspect of this topic analyzed concerns the possibility of limiting citizens' travel. One case study involved a telebiomedical laboratory. It is a medical laboratory in which blood tests are performed both by the patients themselves at home and by biomedical technicians in satellite clinical centers through Internet of Things (IoT) biomedical devices interconnected with Hospital Edge/Cloud systems that allow visualization of the results sent to doctors belonging to the federated hospitals for validation and/or consultation. The relevant issues were addressed with a focus on the need for tele-biomedical laboratories that adopt IoT, Edge, and Cloud technologies.

7.1 Smart Solution for Shopping Guide

In the retail industry, physical stores are turning from merely places to sell products towards spaces for consumption experience. Technology is a key driver behind personal recommendations and other contributors to that experience. The targeted use of technology turns shopping places into networked cyber-physical deployments with the involvement of access points, mobile devices, smart shelves and dispensers, electronic shelf labels (ESL) and point of sales (POS) stations, cameras and other sensors, and industry-specific software such as Enterprise Resource Planning (ERP) for dynamic stock management, label designers, rule engines and campaign dashboards [168]. The more technology is installed, the more options there are for exploiting it in terms of providing a smart shopping experience; yet at the same time, store owners are cost-conscious and prefer low-cost, low-maintenance solutions. Moreover, the added value of the technologisation of stores is not always clear to owners, although it becomes more clear when it translates into higher customer engagement and satisfaction. Letting the customer find the right product with technology assistance, and not missing out an opportunity of purchase, is contributing to the added value as evidenced by recent empirical studies [169, 170].

The focus in this paper is specifically to investigate the feasibility of cost-effective augmented product search within the stores. Consumers interested in fully defined products, brands or less defined categories need assistance in expressing their interests, getting an overview situation about the availability, and receiving guidance through signalling and navigation to ensure that the chosen products end up in the basket. From a technology perspective, this requires a complex workflow encompassing an interactive device (usually the consumer’s mobile phone), beacons for indoor positioning, and shelf labels associated to products. Building an unconstrained lab-level technology could address this problem but would not have chances of being adopted on the market. Instead, we opt for a real-tech approach, intending to design and validate a solution that works in the constrained environments found in real stores and matches real cost requirements. The key contributions in this space are: (i) An abstract workflow for personalised and privacy-preserving physical product search in a shop, (ii) a concrete mobile application to realise this search with hybrid notifications based on plausible technologies, and (iii) a validation in a lab environment with real-tech equipment and processes.

7.1.1 Related Works

Research on digital shopping assistance especially around navigation to products in the physical space has been a niche topic for a long time but has seen progress in the recent years. For visually impaired people, assistance is obligatory and can be addressed with autonomous navigation based on computer vision, text recognition and text synthesis [171] as well as combinations of computer vision and barcode detection [172] and the combined use of accelerometer, gyroscope and magnetometer [173]. The effectiveness of search increases with the data and suitable data structure modelling and visualisation, and therefore research has also been conducted on taxonomies and ontologies such as OntoNavShop [174].

Researchers have also investigated the use of connected devices in shops for other purposes beyond impairment such as smart shopping carts that follows the consumer autonomously [175], technology-enabled personalisation (TEP) [176], and the effects of using mobile devices with augmented reality on consumer behaviour [177]. A previous work studied product-awareness shopping through RFID [178] but required the consumer to be already close to the product to retrieve its information. Many of the studies are conducted with an economics background and do not dive deep into technical matters of feasibility and realisation. In contrast, our work combines physical product finding and hybrid notification about products of interest, an aspect lacking from many of the proposed approaches, and establishes a technological grounding. The hybrid notifications exploit the growing deployment of electronic shelf labels, a technology already investigated from a psychological perspective in terms of revenue effects [179] and customer acceptance [180] but not yet in the context of navigation.

From an innovation perspective beyond the research, indoor navigation and physical product search is increasingly commercialised by startups such as MobiDev and Hyper, and attracting the interest of large mobile platform operators and advertisement brokers such as Apple and Google.

7.1.2 Preliminaries

Localisation of moving entities, such as customers in a store, is possible with multiple techniques. Recent research reports about a precision of around 2 cm that can be achieved with a high number of Ultra-Wide Band (UWB) nodes, for instance [181]. High deployment cost, low mobile device adoption and less stringent application requirement however lead to more balanced decisions on localisation technologies. Moreover, privacy concerns have been raised in the camera-based first smart shopping discussions [182], even leading to broad media coverage[1], leading to further trade-offs. QR codes alleviate these concerns but require active scanning, similar to NFC tags. Bluetooth Low Energy (BLE) beacons are another contender in this space but require dense deployments to achieve tolerable precision and expose a highly device-specific performance [183]. BLE technology is affected by the influence of obstacles [184]. Limitations of precision or acceptance have little effect on the use case analysed in this paper. In our high-level workflow, we do not make any specific assumption and instead merely assume the presence of a suitable localisation subsystem. Table 7.1 gives a high-level indication of advantages and disadvantages of the main method families.

Method	Indoor	Cost	Precision
GSM tracking	yes	high	low
Camera tracking	yes	high	med-good
GPS/GNSS	no	low	medium
BLE beacons	yes	medium	med-good
BLE AP	yes	high	medium
UWB	yes	high	good
QR codes, NFC	yes	low	–

Table 7.1: Localisation methods and technologies

Similar to the localisation, there is an open design space concerning the notification channels for searching as well as guiding and navigating users. Technologies should be inclusive, not requiring any particular device (assuming the search could be initiated with a kiosk at the entrance or via a service robot), and be of low cost from the store owner perspective. The corresponding overview is given in Table 7.2. It is evident that using the personal mobile phone has the advantage of supporting both visual and audible notifications. Electronic shelf labels (ESLs) are less intrusive and, despite having a certain installation and maintenance cost as well as potential security challenges [185], can be a suitable choice if already installed especially due to their proximity to the products. Again, our workflow abstracts from the possible notification options and only assumes the presence of at least one.

[1] e.g. Swiss railways shops `https://awiebe.org/en/sbb-uses-cameras-for-facial-recognition/`

Method	Inclusive	Cost
Mobile phone	no	low
Mobile scanner	yes	high
Earplugs	no	low
ESL LED	yes	medium (battery)
ESL pageflip	yes	low
Kiosk screen	yes	high

Table 7.2: Notification methods and technologies

7.1.3 Personalised Shopping Workflow

This section describes our first contribution, the workflow that allows customers to search for a product in the shop. The workflow shall be characterised by combining personalisation, i.e. considering the consumer's search preferences, and privacy preservation, i.e. allowing anonymous use. These characteristics furthermore relate to the coupling of search and notification through temporarily assigned numbers or colours, depending on the notification channel, in order to support multiple concurrent physical product search activities within a store.

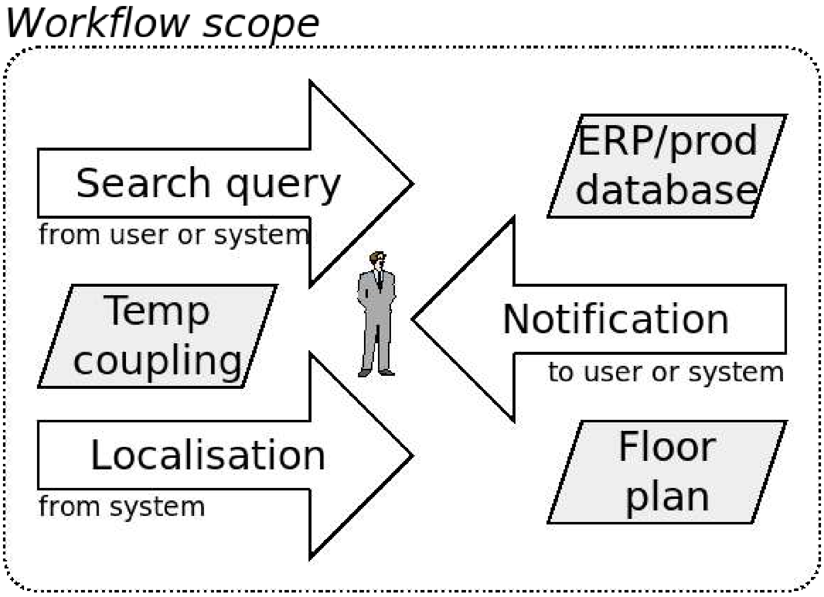


Figure 7.1: Scope for the personalised shopping workflow.

In conjunction with the various options for localisation and notification, the scoping of the workflow is determined according to Figure 7.1. It connects the three main activities with the necessary data structures, indicating an initial data curation effort by the store owner which can however draw on what stores using shelf labels already have, thus not causing additional cost related to the input data.

Referring to the detailed workflow specification expressed as sequence diagram in Figure

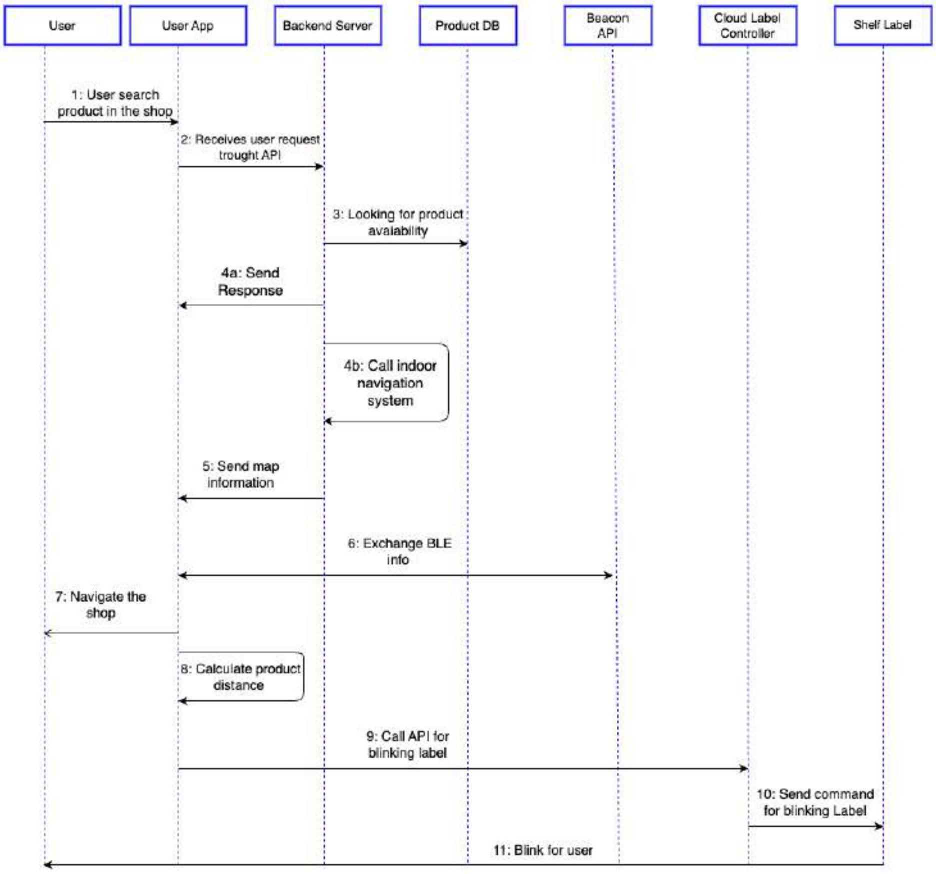

User
User App
Backend Server
Product DB
Beacon API
Cloud Label Controller
Shelf Label
1: User search product in the shop
2: Receives user request trought API
3: Looking for product avaiability
4a: Send Response
4b: Call indoor navigation system
5: Send map information
6: Exchange BLE info
7: Navigate the shop
8: Calculate product distance
9: Call API for blinking label
10: Send command for blinking Label
11: Blink for user

Figure 7.2: Sequence diagram of indoor consumer interaction.

7.2, each phase will be described with greater attention to the most innovative technological components. The workflow either starts from the consumer who desires to search for a product in the shop, or by the system upon the consumer entering the shop with previous preferences saved. For simplicity, we focus on the first variant (Step 1 - User search product in the shop). The user types a text reference of the product (possibly using voice recognitition and speech-to-text conversion) and chooses the one that interests him/her from the list of available products, or a set of products matching a desired category. Again, for simplicity, we focus on the single product search case. At this point the application sends the data with the searched product to the Backend server (Step 2 - Receives user request through API). The server checks the availability of the product in the database (Step 3 - Looking for product availability). The database is updated by the shop owner or automatically from the shop management system. The Backend Server responds to the User App with a positive or negative ack of the research (Step 4a - Send Response). If successful, it returns the position of the product and some information. In parallel, the indoor navigation algorithm calculates the initial route to reach the product (Step 4b - Call indoor navigation system). Map and navigation information is then sent to the User App (Step 5 - Send map information).

The User App communicates via API with the positioning devices installed in the store; again for simplification, we refer to one option, BLE beacons (Step 6 - Exchange BLE info). The data exchange allows the indoor navigation algorithm to guide the user towards the shelves with the product (Step 7 - Navigate the shop). The User App is updated indicating the distance from the product which is recalculated during navigation (Step 8 - Calculate product distance). When the user arrives within "visibility" distance of the labels on the shelves, the User App via API passes the information to the Cloud Label Controller (Step 9 - Call API for blinking label). This component knows the position of the labels for each product. Moreover, depending on the chosen notification method, it is aware of the assigned color of the LED or number on the flipped label itself that the user expects to see. If the label exists and is working, the Cloud Label Controller sends a command to the label to make it flash or pageflip (Step 10 - Send command for blinking label). At this point, the label containing the information on the product sought flashes with a specific color or number which will be recognised by the User (Step 11 - Blink for user). The workflow described allows a user who is looking for a specific product in a shop to check its characteristics and availability and search for it on the shelves without wasting time physically searching for it. The flexibility of the workflow opens up the possibility of future developments that can, for example, suggest a product to the user on the basis of a profiling process and allow him/her to reach it, take it and paying the product directly in the app.

7.1.4 Mobile Interaction and Navigation

This part of the section discusses customers' interaction with the mobile application and underlying technologies. For the purpose of a better understanding of the technologies, the case of the search for a single product is reported. The functionality can be extended to a list of products. The mobile application is designed for the user's smartphone. Nowadays many people use smartphones on a daily basis. These devices are designed to work with different technologies including BLE-optimising battery consumption. There are three subsections through this section: Locate User, the part where the navigation process takes place; Search Product, where the customers search for a specific product they want to buy; and Navigate to Product, detailed information about the searched product. Each of those refers to a subprocess from the previously explained workflow, providing a concrete realisation for the mobile device side while remaining flexible for the infrastructure side in terms of beacons and ESLs. The interaction-centric discussion is based on a distributed software architecture connecting the necessary system components for search, localisation and notification as shown in Figure

10.1.

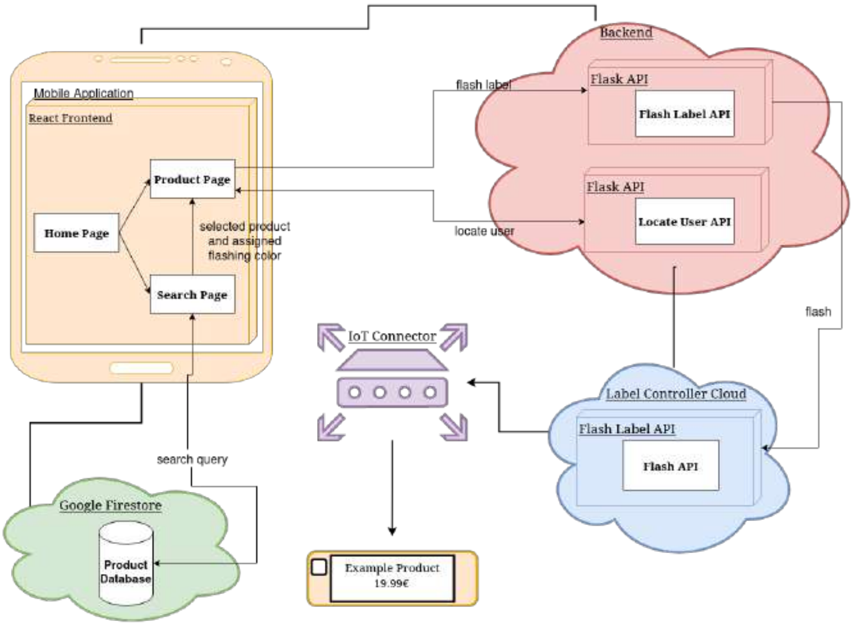


Figure 7.3: Mobile application architecture.

Locate User

On the mobile application home screen, there are two main paths on with which customers interact: Locate User and Search Product.

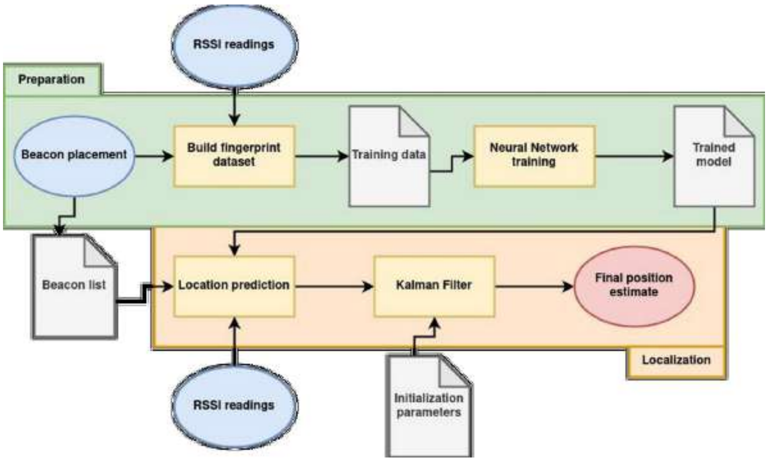


Figure 7.4: Beacon-based navigation algorithm.

On the Locate User path, there is a straightforward process: indoor navigation using BLE and locating the customer on the floor map of the shopping store. Depending on the physical deployment, the Bluetooth signals may arrive from a ceiling-mounted access point; in this case, either a single AP provides angle-of-arrival support to determine the direction (and the user's mobile device supports the necessary BLE protocol version), or multiple APs are used for trilateration. Alternatively, if no AP is available or does not provide a suitable API, a mesh of BLE beacons can be deployed, calibrated and used for the same purpose, with configurable density to balance deployment cost and localisation precision. In our implementation, based on existing research we provide a Neural Network-based navigation algorithm which is competitive in accuracy. Combination of Bluetooth fingerprinting, a Neural Network, and a Kalman filter to predict the position of a user is used for the navigation, as expressed in Figure 7.4. The algorithm is separated into two phases, which we refer to as the preparation and localization phases. For the preparation phase, training data is collected by moving a Bluetooth receiver device between as many different points on the shop floor as possible and collection signal strength (RSSI) measurements from the BLE beacons. This data forms our BLE fingerprint database and serves as the training data for a feed-forward neural network. It should be noted that by conventional terms this model is over-fitted, as all the training data is collected from the same location and so it would not work in a different location unless retrained. This is however the state-of-the-art in neural network-based fingerprint localisation, and the traditional alternative of multilateration based on the RSSI measurements [186] also requires manual calibration on location. The model can then predict the location of the receiver (a user's smartphone) and the prediction passes through a Kalman filter for smoothing in between measurements to reduce the jittering of the position the user sees on their screen.

Search Product

The Search Product path in the application is designed for sending search queries to the database where all the products are stored, typically an ERP, but alternatively a Firestore database with generic schema that works out of the box in our implementation. Customers can easily search for any products they want on this page and then connect to the localisation to correlate both the customer position and the product position. In addition to the search query feature of this page, the other important function is the personalised assignment of an anonymised results indicator, in the form of a colour or number related to the notification channel.

The ESLs available on the market and used in that research have limited flashing colours for flashing commands, and limited preloaded e-ink pages for pageflipping. To avoid customer confusion, each customer should have a unique flashing colour or display number to track the ELSs applying to the appropriate search results. Nevertheless, it is impossible to assign unique colours or numbers to each customer with the current hardware technology. Colours are usually limited to single-digit amounts, and e-ink pages to low double-digit amounts. Therefore, each customer will be assigned different colours or numbers temporarily during the product search, with the mobile application informing about the assignment. If all possible colours or numbers are occupied, customers will be informed and move to the standby list if they wish. Other possible approaches to increase the physical notification options beyond the phone itself are possible, but not currently implemented by us, such as combinations of colours and numbers, or different blinking LED frequencies or patterns.

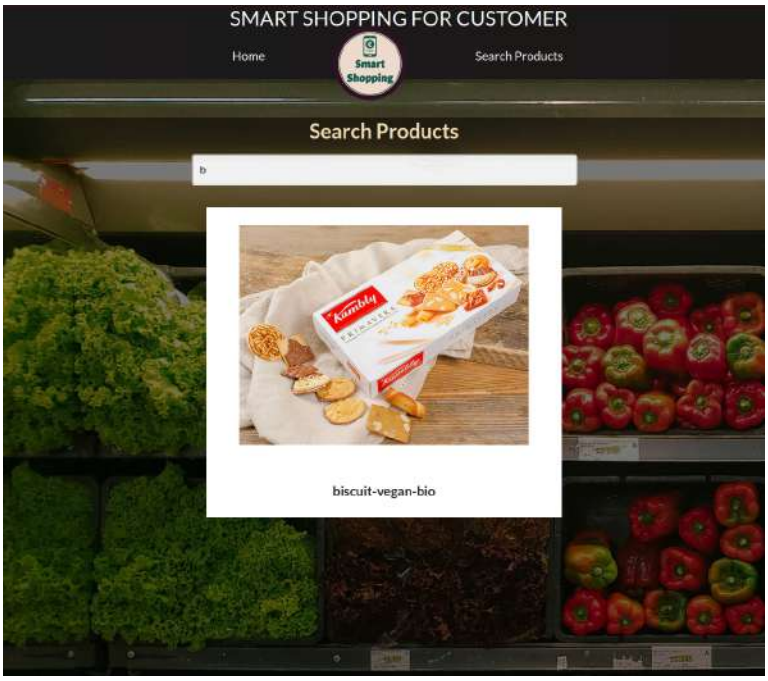


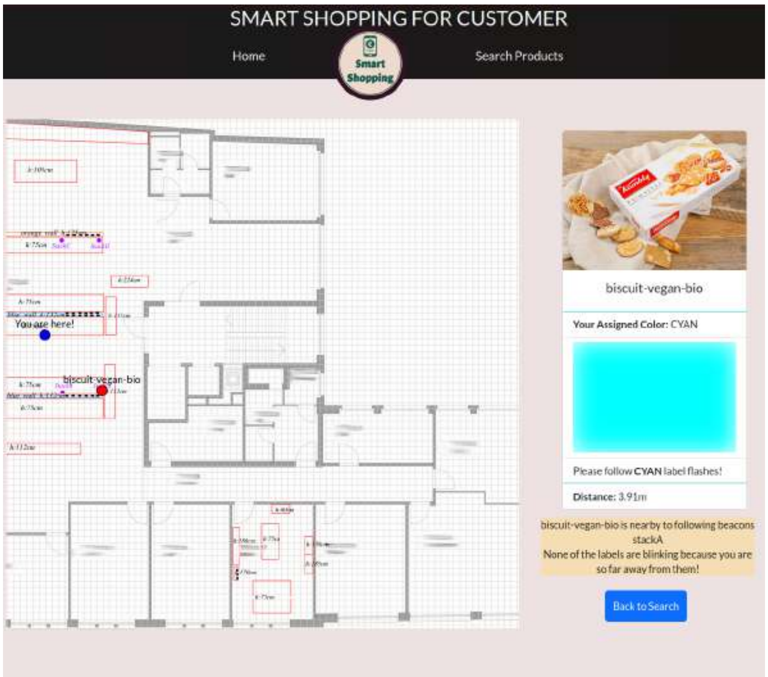
Figure 7.5: Search Product sample view.

A sample view of the Search Product entry page for a hypothetic store associated to our physical research lab premises, as outlined in the validation section below, is shown in Figure 7.5.

Navigate to Product

Customers will arrive at the Navigate to Product Page if they search for a product and click that product on the Search Product page. The Navigate to Product step is the final yet potentially longer-run destination of the customer. Here, customers can find information about the product, such as product location on the floor map, distance from the product, price, and a descriptive image. In case no assignment was performed yet, the assigned colour or number will be first displayed on the Navigate to the Product page. The same assignment is then shown as a reminder on the Navigate to the Product page. In case of all colors are occupied, customers will see that in the pop-up screen, and if they wish, they will move to the standby list until a color becomes available. Even without assigned colour, the map-based navigation on the mobile device itself provides a suitable fallback, although it excludes customers without a phone or without the application installed.

Again, a sample view of this subprocess is provided in Figure 7.6. It shows the floor map on the left side, with an overlay for navigation consisting of two to three main items of information: The current location of the customer, the location(s) of the product(s) resulting from the search, and possibly, although not presently implemented by us, a preferred path to collect all products, for instance based on the shortest path navigation. The addition of the path would be more useful in practice in larger stores or malls. On the right side, the page shows the next product in the results list along with navigation information and the assigned personalised indicator.



biscuit-vegan-bio

Your Assigned Color: CYAN

Please follow **CYAN** label flashes!

Distance: 3.91m

biscuit-vegan-bio is nearby to following beacons
stackA
None of the labels are blinking because you are so far away from them!

Figure 7.6: Navigate to Product sample view.

7.1.5 Technology Fitting

To determine the feasibility of our approach, we have validated it under realistic conditions, in a research laboratory for smart technologies, following a real-tech approach by using commercial technology widely deployed in stores today as integration points. This concerns especially the ERP to obtain product information, the ESLs, and the label controller to interact with the ESLs. Due to proprietary communication protocols, the tight coupling between ESLs and label controller is unavoidable, whereas the other technological choices permit a degree of flexibility. Table 7.3 contains the details on all chosen integration points. The table also informs about an approximate and rounded price point in € in order to facilitate the discussion on how economic the resulting solution can be especially for smaller stores. Of particular research interest in this context is the ability to replace existing functionality with an open source implementation that can be used to foster innovation. This analysis provides our third contribution.

Category	Solution	Cost
ERP	ExtendaGo	100 €/y
Label designer	Vusion Studio	400 €/y
Label controller	Vusion Optipick	350 €/y
Self-localisation	Mist API	300 €/y
Self-localisation	*our approach*	–
Mobile application	*our apporach*	–

Table 7.3: Integration points (APIs, portals) for end-to-end validation and comparison

Experiment Testbed

Our testbed setup resembles a small store with three longer shelves, a total of 30 ESLs in use to mark products, and 12 BLE beacons. The one-time hardware cost is around 200 € for the ESLs, 200 € for the IoT adapter, 700 € for the AP and 120 € for the beacons. In order to have greater flexibility for investigating mobile device behavior, we have used a Linux-based notebook instead of a mobile phone to interact with the system.

By interacting with the ERP and generating label images dynamically on our backend system, both for the product and the numbered pageflip pages, we are in a position to discard the label designer. Moreover, by being able to tap into beacon-based positioning, we are also able to discard the existing localization API. Store owners who prefer to use those online platforms will still be able to do so with our implementation.

We set up the products on the shelves with a 1:1 mapping to ESLs. For labels that emit BLE signals, these could be used as a high-density grid for the navigation. However, most products on the market do not emit signals. Therefore, in our testbed, we assume one beacon per running shelf meter, with the aim to lure customers nearby the target shelf area. Once nearby, the local notifications such as label flashing and pageflipping can occur. An impression of the testbed is given in Fig. 7.7.

In the experiment environment, 12 BLE beacons were positioned in the research lab space

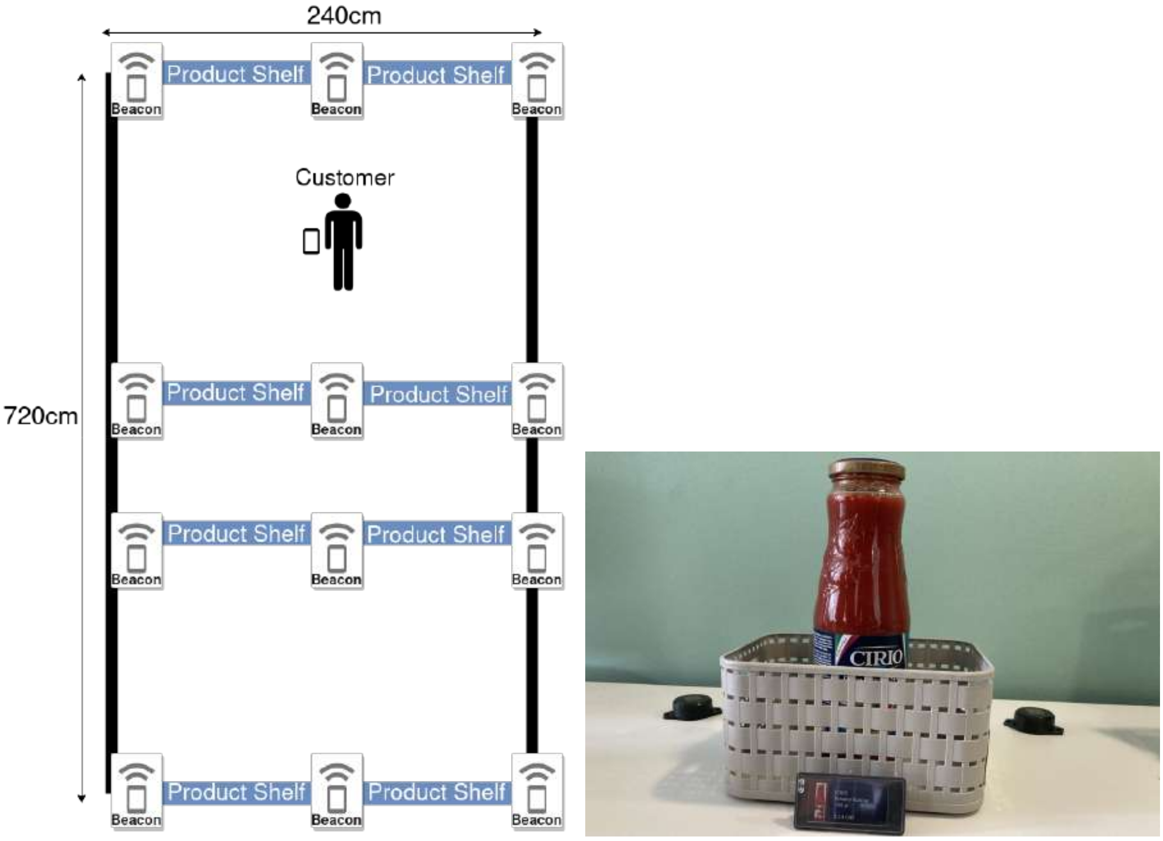


Figure 7.7: Schematic view of physical experiment layout, and impression of a product.

at specific places to create a 4m x 7.2m grid spaced by 2m representing the shopping store along room dividers representing the shelves. The schematic grid is shown in Figure 7.7, left side. RSSI measurements were then collected at different positions to collect a fingerprint dataset to train the Neural Network model. The dataset was used as training data to generate a Neural Network model that predicts customers' positions based on RSSI readings. The RSSI readings are collected in the background from the 12 beacons in the grid every 4 seconds. The frequency of 4 seconds was selected to ensure all beacons are given the chance to advertise and be received by the mobile device. The neural network inference itself is actually much faster. Since that technology will be used in shopping stores, the Neural Network model is purposefully overfitted to get more accurate predictions, meaning the model is only usable in the location it is trained in. Once the model is trained, it is used to predict a position which then passes through a Kalman filter to smoothen the value and rule out spikes. The output of the filter is the final output exposed by the localisation service. The final result represents the location of the customer on the floor map.

Software Implementation

The implementation of the web application used for validation purposes was built using several underlying technologies. For the front end, React JS was used to create the user interface. React JS is a widely used JavaScript library for building user interfaces and allows for the efficient and scalable development of complex web applications. The back-end service was built using Flask API, a microweb framework written in Python. Flask API allowed for the creation of RESTful APIs that could be used to interact with physical devices (label controller for ESLs, beacon/WiFi scanner). Requests sent through the React front-end were able to interact with these APIs to retrieve data from and send data to the IoT devices. Finally, the Google Firebase platform was used as the database for the web application. Firebase is a cloud-based database service that provides real-time updates, secure user authentication, and scalability, making it a reliable and effective choice for the web application's database needs.

The software implementation resulting from our research is available as open source [187]. We expect that it helps accelerating the setup of real-labs for personalised shopping and product search in the future.

Experiment Findings

Our findings cover both economic and technological considerations. Concerning the localisation, we can confirm that indoor navigation based on beacons is feasible for a shop environment concerning the navigation precision towards an area close to the target shelf, and that despite additional investment, the overall cost may be lower if such a deployment is planned from the start.

The mean positioning error of the model is typically 50–100cm (the resolution of the grid), although spikes occasionally occur (Step 6 - Exchange BLE info and Step 7 - Navigate the shop). The accuracy is generally higher than RSSI-based multilateration as used in the state-of-the-art and comparable with more advanced solutions such as angle-of-arrival-based detection [188] or UWB [181] which is more expensive to install and also incompatible with the majority of customer smartphones. With further training rounds, which could be automated by piggy-backing on cleaning or restocking robots in stores, the precision can be expected to increase slightly.

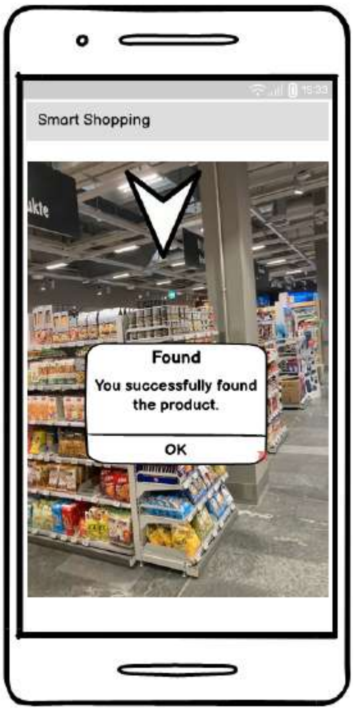
Assuming a write-off period of five years for personalised shopping equipment, store owners today will have to invest 6850 € (including all hardware except for the beacons) to get production-grade support for introducing ESLs and for being able to access a raw positioning API. At this cost, they would still need an on-top solution for personalised navigation and physical product search. In contrast, our solution requires the beacons but is able to discard two existing platforms and the AP as mentioned above. If WiFi is required in the store, a more reasonably priced AP could replace it, with a presumed cost of 200 €. This results in a total cost of 2970 €, equivalent to 43% of the comparative investment, and with the added benefit of obtaining an integrated solution for search and navigation.

7.1.6 Final Remarks

With our work, we have achieved to demonstrate technical and economic feasibility of physical product search and navigation to these products in stores. Through conscious technological choices, our result has proven to work in a real-tech environment, can be implemented with low-cost hardware and a minimum set of online platforms, and works with low power consumption. For the customers in the store, a privacy-preserving and inclusive experience is provided, increasing the likelihood to boost sales and revive physical shopping across target populations such as digital natives, elderly people or tourists. As a tangible result of our work, we have published an open source software implementation [187].

Our research has focused on support for hybrid notifications including navigation on the mobile device. From a human-computer interaction perspective, additional modalities to receive nagitation advice and notifications could be built on top of our work. This includes augmented reality (AR) navigation to maintain the overview in larger shops with multiple separate shelves hosting the desired products. Our implementation is prepared for this modality and we are in the progress of building the first AR-based navigation as sketched in Figure 7.8.

This research moreover leads to a unique economic value proposition. According to our interation with store owners, especially for high-value stores there is a high need for such a solution. The economic potential is therefore in commercialising the research results. If a price tag of around 8000 € is aimed at, only slightly above the current mark but with better functionality, this would imply a possible range of more than 5000 €, divided into both the development cost and sales profits. In our follow-up work, we will therefore focus on the following directions. First, we will study the performance and price point for AR-based navigation. Second, we will investigate the scalability of the solution in larger stores, and the operational/maintenance perspective including the adoption of more sustainable technology



Found

You successfully found the product.

Figure 7.8: Ongoing augmented reality integration prototype.

such as solar-powered ESLs and beacons that are technically suitable for indoor shopping lighting conditions. As a third research direction, in order to support both customer and shop owner we will investigate additional features. For that matter, we expect an emerging mobile application to incorporate cutting-edge technology that enables users to search for a product and receive recommendations for related items. Once the customer has added all desired products to their basket, the application will generate the shortest walking path from their current location to the cash service, including all products in the route. The path precision will be increased by leveraging multi-sensor fusion and multi-perspective consensus voting [189]. Along this path, the application will suggest additional products nearby to the customer as they navigate through the market. If the customer accepts any of the recommended products, the application will automatically generate a new walking path with the same logic. This feature will provide a seamless shopping experience for users, helping them discover new products while efficiently navigating through the store.

7.2 Innovative Methods for Indoor Navigation

In the digital age, social networks are widely used for sharing news, opinions and information related to the physical place just visited by a user. In this way, through the use of innovative tools offered by social media, it is possible to geotag a place and share your opinions with the connected world. One of the main problems we encounter when using

Social Networks is "Trust". A user can share an opinion, a post without there being any form of control over the veracity of the information he or she publishes. The data generated and shared by users is a very important source of big data, as the information contained in the geotags is always updated to the most recent state and, by making a comparison on their popularity it is possible to define a relationship of trust with them. Visiting a place using only social networks with information shared in geotags turns out to be an incomplete experience. But merely reading information and posts does not allow us to have a real and complete vision of the environment of interest. This technological limit can be overcome by integrating the most advanced Virtual Reality techniques with social media. Virtual Reality is currently a technology that is little present in people's daily lives, but it is used as a support for training and virtualization in many contexts such as Industry 4.0, medical applications, education, cultural heritage, the real estate market, in tourism, etc. In this article we want to propose a solution to the limits of social networks, integrating virtual reality as a tool applied to discover, explore, visit any place reproducible within a virtual environment through a 3D engine [190]. Visiting a faithful virtual representation of an environment allows users to have a more realistic experience than viewing a comment or tag on a web page. Furthermore, in the virtual visit, it is possible to insert information that is not contained within the Geotag and which can support users in choosing whether or not to visit a place in reality. In recent years, the new frontier of digitalization passes through BIM applied to existing buildings, in order to consolidate the concept of Digital Twins [191]. BIM is a technology created for the design of new buildings and or the digital reconstruction of existing buildings characterized by the integration of information and metadata that enrich the structural model. In particular, Heritage Buildings Information Modeling (HBIM) uses the latest photogrammetry and laser scanning technologies to create a three-dimensional virtual model faithful to reality. By integrating BIM models [192] into an Engine, users can interact with a simulated parametric environment. From BIM it is possible to extract the virtual 3D model and create a software infrastructure capable of managing all the "geotagable" geometric and structural information. Within the new perspective of digital twins, the choice to integrate two consolidated technologies such as BIM and VR can give us the possibility of defining an interactive tool through which it is possible to have a virtual overview of the state and information associated with a real building. With this work we want to propose an innovative platform that integrates social networks with new technologies such as VR, BIM and IoT in order to define new discovery features during virtual navigation. Our idea is to offer an innovative method for finding and creating geotags. In general, the information relating to a place visited directly through modern "reality" simulators faithfully reproduces the place of interest. In order to achieve this goal, we will apply different technologies that can offer different types of services to end users in a single platform. Through this work we want to define a new method for referring to the world, in such a way as to geohash the hierarchical and three-dimensional code based on the Open Location Code algorithm. These codes will be used within the geotags to point to areas of different (depending on the length of the geocodes). Next it will be necessary to develop a platform capable of enriching and recovering from all types of geotags. In particular, the platform will store:

- Internal geotags created by the user of our platform;
- External geotags retrieved from Twitter and Flickr;
- IoT geotags, which are the data generated by IoT devices installed in the reference area.

Within the platform it is necessary to integrate these technologies with a virtual 3D building, while for virtual reproduction we will use the BIM model. The choice to integrate the BIM model into the Engine was made because BIM:

1. it is an innovative technology used for the digital and parametric representation of a building;
2. allows you to integrate authoritative geotags (i.e. information that can be read as geotags) with your infrastructure;
3. its layered structure allows you to insert trusted internal geotags at any level of the structure;
4. Can be easily integrated into the navigation engine engine.

To offer the user a realistic and immersive virtual experience, we will use the 3D Unity Engine, which allows you to import BIM models [193] from Autocad software and make them interactive. In particular, within Unity Engine, we will retrieve all the geotags stored in our platform and show the user the correct location to discover the place that users have tagged.

7.2.1 State of the Art

The impact and rapid diffusion of Social Networks in today’s society has meant that this new phenomenon has become a primary source of Big Data, easily interrogatable, interacting and constantly evolving. Compared to the multiple data sources present in different contexts, Social data are localized, and the content of the localization has been subject to multiple analyzes in different scientific works, especially for marketing purposes. In particular, in [194] the authors used geolocalized multimedia content downloaded from Flickr and Panoramio to understand which are the most important points of interest (POI) in a city. A similar work was proposed by the authors in [195], who, tried to carry out a differentiation of POIs in a city for local and foreign visitors using the geolocalized posts on Twitter and Flick of the visitors to understand which information is valuable and which one doesn't. These works exploit information from people's data, but consider only a part of the entire existing data. Data from social network sources such as Twitter, Flickr [196] and so on are more similar to outlier data than average data, this occurs because only part of the data is localized. This thesis is partly justified in [197] where the authors try to understand who enables geolocation services in Twitter, and the demographic differences between users who publish a post. As can be seen in the literature, there is a clear and growing distrust in geotags, as they are considered as a reliable source of data useful for understanding human behaviour. For this reason, many recent studies are focusing on changing the ways in which this data is shown, contextualizing it in maps, videos, virtual or augmented reality applications. In this way, the data is shown to users according to appropriate and direct localization criteria. In [198] the authors used a simple 2D map as a basis to show tweets localized in relative areas. Furthermore, in [199] Social Street View is proposed, a solution for the visualization of geodata on a geographical context. In particular, a Street View map is enriched with geotags from sources such as Instagram and Twitter. In this work it was possible to analyze the union between the concept of map exploration and tag exploration, but without immersive results. In [200] the authors proposed Geollery, the most advanced combination of geotagging and "trendy" cartographic exploration of social networks. Authors can build entire real-world environments such as cities, parks, etc. using real maps, while allowing users to navigate in VR the 3D virtual reconstruction of a real environment. In this work the reconstruction of virtual places is not faithful, and indoor navigation is impossible. To define an accurate representation of the real environment and buildings, and at the same time integrate it with geotags, an optimal solution can be the use of BIM models. Having an interactive virtual environment proves to be a challenge that has been extensively reviewed in the literature. For the reconstruction of a real environment, methods are applied for the virtualization of existing buildings which allows converting a building acquired by laser scanning into a 3D model on a BIM platform [201]. In particular, this tool allows you to obtain the point cloud used to create a 3D model based on a mesh. The reconstructed BIM model is combined with the latest virtualization experiences to develop semi-immersive applications in Augmented Reality (AR). In [202] the digital content is superimposed on the geometric model in order to define an interaction with the real physical space. VR technology can be combined with Geographic Information System (GIS) and BIM simultaneously to solve problems in building Sponge City [203]. With BIM it is possible to detect design (construction) and management problems of large areas and reduce construction costs by improving efficiency. Starting from a preliminary analysis of BIM technology and its integration with VR, we can identify several innovative applications. In [204] the authors summarize the studies of the Post Occupancy Evaluation (POE) method to provide a detailed framework capable of integrating the collected data and 3D spaces all the information and at the same time define easy-to-use functions and provide tests and tools data visualization. In this work, we want to put together the concept analyzed so far to propose a tool, which, through the innovation of geotags and BIM models, allows you to navigate in an enriched virtual reality to give users the opportunity to live an immersive experience.

7.2.2 Motivation

The proposed study is based on the need to use geotags to know and discover real virtualized places. From the analysis of the state of the art, it emerged that geotags are used in different fields such as tourism, marketing, advertising, etc. In particular, they can be used to show the POIs of a city, or to understand the performance of a product and/or market, or simply to share an opinion or idea. Geotags are a technology with enormous potential, numerous, always updated and decentralized, but they are currently not considered a reliable tool. Basically there are two reasons that affect the reliability of geotags:

1. They are outliers in that only a small fraction of people in the world use geotags, and therefore only the opinion of a small percentage of people is shared;
2. their position is not guaranteed: in Twitter, for example, it is possible to enable the position, and in each tweet insert the position we were in at the time of publication. In cases where we post an event that happened on the other side of the world, there is a loss of "trust" in the post.

These problems have been widely analyzed by the scientific community, in particular in many works geotags have been considered a source of Big Data from which to extract information for the definition of sophisticated AI (Artificial Intelligence) algorithms. Careful analysis shows a lack of focus on the display of geotags by end users, who must be able to decide how to use the data contained within them. To solve the visualization problem, many scholars have proposed the insertion of geotags within a 2D map, or being contextualized in virtual social networks such as Social Street View and Geollery, as a solution. In particular, users are given the opportunity to move within a virtual map and explore the reproduced environment using geotags. Many works introduce internal navigation with geotags, while other solutions see the geotag as a central element in external environments such as squares, streets, etc. To be able to use geotags in closed environments such as theatres, public offices, museums, etc. it is necessary to integrate social media with sophisticated virtual reality technologies. Using virtual reality and integrating a structured model like BIM into a simulated environment gives users the ability to navigate and explore a digital twin of a real location. The choice of BIM models is made because, in addition to providing faithful information on the geometry and structure of a building, BIM can be seen as a database of geotags of the building modeled in the real world. In a BIM model we can have structural geotags such as the size of doors, the presence of architectural barriers and the slope of a staircase. Social geotags can be integrated into this type of environment, so as to enrich BIM models. In the future of buildings connected via IoT devices, within the scenario of Smart Buildings and Smart Cities, the amount of data collected can be shown to users using geotags in VR and enriching them with information added by other users, such as opinions and reviews. With this work we want to integrate these technologies into a single platform with the aim of bridging the gap currently existing in the state of the art of Geotag Living and indoor navigation. In fact, we think we can create a system capable of aiming to solve various objectives that are not being achieved at all at the moment, such as:

- the creation of a self-trusted geotag database;
- the use of digital twin buildings as a source of geotags;
- the creation of a geotagged indoor navigation system;

With this work we do not simply want to improve the state of the art on virtual navigation and the geotagging exploration system, but we want to lay the foundations for a future change in the way in which people explore and experience the world through technologies that already they know.

7.2.3 Platform Design

The technological system designed is based on a microservices architecture, and the different components have been designed to carry out specific functions:

- The Geocode algorithm is used to "represent" the location;
- The geotag structure must be created to organize information within a Big Data context. A weak structure can lead to multiple problems, as reported in the State of the Art;
- The Virtual Environment must integrate the BIM model with VR functionalities in such a way as to make the 3D model interactive and completely navigable;
- The geotagging platform must collect, store and analyze geotags;
- The platform frontend must be able to make the user interact with the geotag navigation system.

The platform proposed as a result of the study carried out aims to give the user the possibility of interacting with a virtual environment enriched with geometric and parametric information of a building. The prototype was created by importing a BIM model of a building onto an Engine. Furthermore, by integrating the geotags infrastructure with the virtual model, the user can obtain information such as opinions and reviews from other users who have previously visited the place of interest.

Geolocation

Classic geolocalization methods are not based on an efficient user experience, and therefore it is necessary to design a geolocalization algorithm capable of guaranteeing precision, simplicity and which can at the same time be a user friendly system. In general, a "geocode" is an alphanumeric code that represents a geographic space, and its characteristics depend on the algorithm used to implement it. Different algorithms often produce different codes and can rarely be combined. Real geotags produced by Twitter and Flickr, such as Plus Code, have a limitation due to the lack of localization of the altitude related to the information produced. For this reason we can label an area on a surface, but not inside a three-dimensional building. This is a limitation that we try to resolve using a three-dimensional geocoding algorithm called the World Domain Name System (WDNS). WDNS is designed according to the Open Location Code [205] which uses clear encoding of latitude and longitude information based on the WGS84 standard. The algorithm works offline and does not require any external configuration. Twitter, Facebook and other social networks are used by a very large number of users but lack accuracy in location functions. In particular, the goal of this work is to have a simple but precise geotag generation process. To this end, it is necessary to define a fundamental characteristic, the precision in sending/publishing data. The geotag must contain a body that collects the thought we want to express, and possibly multimedia content such as photos, videos or audio tracks. Furthermore, it is necessary to define a content type, which can be a string, URI, whose value allows us to understand how to manage the body [206]. Therefore, an information type field helps to understand the "context" of the data. One type of information could be the value that one of these example fields can take: opinion, public service, government information, and so on. Finally, the “location” field is an explicit location of the data we want to label, and of course, the value is a WDNS code.

Virtualization Process

Building Information Modeling (BIM) is the holistic process underlying digital transformation in the architecture, engineering and construction (AEC) industry. Furthermore, the AEC industry is currently studying strategies to make the buildings of the future smarter, more efficient and more resilient. In general, BIM can be defined as a digital model that represents the geometric, physical, material and economic aspects connected to a building. In reality, this technology is used to create and manage all workflows and data in the life cycle of a construction project. The innovation of BIM models consists in the interconnection of digital information in open data and protected data. This data can be extracted and used in different applications in order to give users the opportunity to have all the information provided by the model itself. Through the interconnected information it is possible to create a virtual model of the buildings, with a data-set, and a data visualization function that gives the possibility to define the analysis of the environment in real time. Furthermore, the BIM database can be integrated with scripts and functionality to create a prototype for rendering with the help of an Engine. VR technology allows users to have a realistic perception of the reproduced environment. Nowadays we are interfacing with the connection between BIM and the new concept of digital twin, both from a conceptual and practical point of view. The transition process from the physical object to its digital replica consists in the reproduction of a 3D Virtual Model of an existing building, obtained with various digital survey techniques based on laser scanners and point of cloud which contain all the information necessary for maintenance and the management of a building. Thanks to Building Information Modeling it is possible to structure data in such a way as to create a virtual replica of a building and connect it with sensors or Geotags. In the case of existing buildings, we are in the presence of Heritage Building Information Modeling (HBIM), a process that increases the potential of BIM models by introducing intelligent and parametric objects with well-defined semantics. With this study, we want to create a smart and optimized 3D interactive model that virtually reproduces the reconstructed environment. In particular, through the use of tools such as Revit by Authodesk, Unity Engine and appropriately designed scripts it is possible to navigate in an existing building enriched with various information associated with Geotags.

Building Geo-localization

The building models that we intend to navigate within our simulation engine must be localized in space as the geotags must be visible in both virtual reality and physical reality. To this end it is necessary to know the equivalent position of the world when we virtually navigate a building. The WDNS is a point that must correspond both in the reconstructed model and in the real, physical position of the building. From the users' point of view, they can view the information in both the virtual and real environments and make the same consideration about the place they are visiting, so that the data is correctly contextualized. To locate the model of a building in the real world it is necessary to match the coordinates of the building model (local coordinates) in the virtual environment and the coordinates in the real environment (global coordinates). Assuming that the virtual model is strictly faithful to the real building, it is necessary to convert the global coordinates to X, Y using the equi-rectangular projection, optimized for small areas. Afterwards, once the scale factor is known, it is possible to align the axis of the real model with the axis of the virtual model by applying a transformation matrix.

Geo-tagging Platform

The architectural design underlying the proposed prototype platform allows us to define optimized functions for the recognition of people, positions and above all to collect, archive and display geotags using the structure that we have defined previously, capable of harmonizing data from different sources . At the basis of this study we find the type of data that we want to collect and use within an automated system for the creation of a particular type of geotagging. This data must be collected by means of the previous data structure reconstructed within our applications and via the platform's API (Application Programming Interface). The building models that we intend to navigate within our simulation engine must be exactly localized in the world, because the geotags must be visible in virtual reality and in reality. The interaction between users and the platform occurs at the highest architectural level, and is specifically built at the API level to take advantage of all the features offered by the platform. We think that the building we have reconstructed on our platform should be easy to navigate in a virtual environment, using technologies we are already familiar with, such as web and mobile applications or 3D viewers. It is necessary to integrate the navigation system with the geotag architecture, to correctly show them in the right place within the virtual building, and this can be done through the APIs that the platform exposes. To this end, Engine Unity fits our goals perfectly. Using Unity, you can easily import 3D models from the BIM model and navigate them by designing algorithms optimized for users' virtual navigation. Last, but not least, you can also use an API client to retrieve and send geotags to the geotagging platform.

7.2.4 Implementation

This section describes the implementation of the platform. The description will be based on a high level architecture.

BIM-based Building Modeling

In recent years, researchers have focused on understanding how it is possible to connect two innovative technologies such as BIM and VR. As mentioned previously, we used the Unity Engine to configure the virtual environment. Configuring the BIM model on Unity is necessary in order to export the FBX (filmbox) file and have all the properties of the original Revit file. With this format it is possible to define interoperability between 3D applications and the 3D BIM model. Using Unity we can directly import the fbx file exported from Revit, but in this way we obtain a model without materials and textures. To solve this problem and to import the textures you need to use a third party software. In the literature, the "3ds Max by Autodesk" software is widely used for creating a static mesh and for setting all the textures of the 3D model. In our work we chose to use the "Simlab Composer" software, with the aim of integrating all the textures into the .fbx model to be imported into Unity. Furthermore, through this tool it is possible to import 3D models, create a dynamic view and render the model imported from Revit. After importing the .fbx model processed with SimLab Compose into Unity, you can create 3D geometry in the virtual environment. During the import process, some issues arise related to integrating all parameters of 3D BIM models into Unity. For this reason it is important to define a backend infrastructure that gives us the possibility of setting some properties automatically via REST API protocols. Some properties such as building scale, orientation, color/texture can be changed directly in Unity Engine. To allow users to view the virtual environment and interact with the information obtained from the geotags, we have implemented some software modules that perform the simulation.

Virtual Navigation

The implemented prototype platform offers the user the opportunity to live an immersive experience and navigate the virtual environment. The simulated scene is the georeferenced BIM model of a building. To implement the FPS Controller functions in Unity, with a first-person perspective, it was necessary to configure some setups in the virtual simulation:

- BIM Model: The process of importing the BIM model into the Unity scene requires specific settings. It is important to import the pre-processed .fbx file with the materials and textures. Furthermore, through a particular configuration it is possible to apply the recursive search for textures and to configure the setup which gives us the possibility of making the BIM model physically realistic in terms of gravity and geometric collision. After these procedures, we can have a navigable virtual 3D model, which is a reliable virtual reconstruction of a real building;
- Rigidbody: is the component added to an entity that adds the action of gravity within the Unity Engine Physical System. A Rigidbody object is affected by gravity and reacts to collisions with other objects within a scene. Furthermore, with the use of the Rigidbody scripting API, it is possible to control an object in a physically realistic way;
- Colliders: it is the component that defines the shape of a GameObject with the aim of setting physical collisions. If the Rigid Body and Collider are not added correctly to the simulation model, they will pass through each other. In Unity, there are different types of Colliders, in our simulation we use a spherical collider for the Player game object;
- Character Controller: it is the component used to implement the player in first/third person paying attention to individual characteristic parameters and movement effects. Additionally, to make your character affected by physics, you need to pair your character's controller with the Rigidbody setup;

To implement FPS virtual navigation, we created a gameObject called "Player" with a child object "Camera". Furthermore, to develop the navigation function, it is necessary to set up two software modules that simulate the view and movement of our first-person controller respectively.

Interactive Unity environment - BIM and Backend API

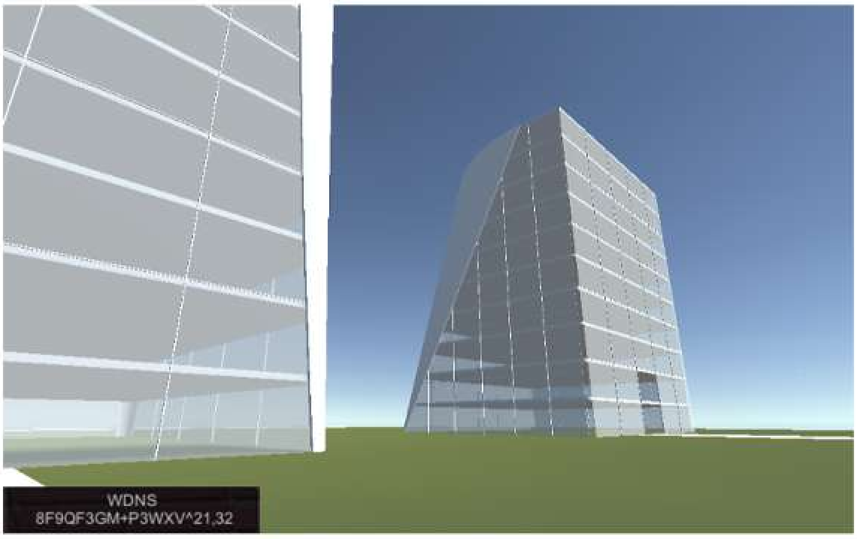
The integration between the virtual model and the Backend system occurs via APIs that allow us to have and provide information on the simulated virtual environment. In particular, through the implemented platform, we can provide two types of information: "geometric" and "reviews". Geometric information, such as door widths, information on architectural barriers, elevators, stairs are extracted directly from the BIM model. Information that changes over time, such as reviews, menus, non-fixed fire exits, offices, etc., is uploaded to the platform directly via geotags. The loading of geometric information into the virtual scene in Unity occurs directly via appropriately configured software modules. While the data associated with the geotags is loaded using the rest API which allows us to extract all the information saved in the backend system. In particular, through the use of the high-level scripting APIs provided by Unity known as High-Level API (HLAPI), it is possible to have access to functions that allow you to:

- manage the network;
- serialize the data through a generic serializer;
- send and receive network messages;
- interact with a client server system;

Furthermore, through the UnityWebRequest library, it is possible to define a modular system for composing requests and managing HTTP responses.

7.2.5 Developed Platform Description

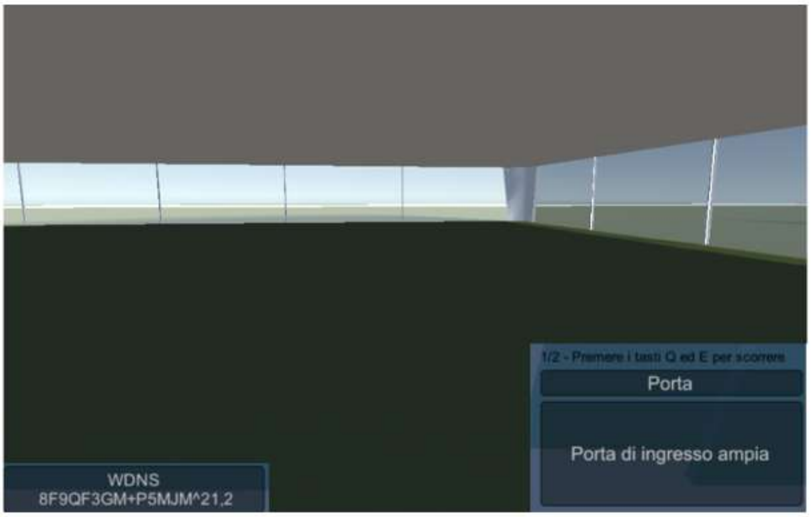
Once the entire platform was designed and implemented, it was tested with the aim of showing its potential and to understand the improvements to be implemented in future work. For this demo, we chose to use a BIM protype model. The front end technology we chose to use is Webgl, exported directly from Unity Engine. Figure 1 7.9 shows a scene reconstructed within Engine Unity. The BIM model, once rendered, is integrated into the scene and navigated through the FPS functions. In the lower left edge there is a box that shows the WDNS relative to the position within the simulation. The WDNS is made up of a



WDNS
8F9QF3GM+P3WXV^21,32

Figure 7.9: Virtual Navigation Demo.

13-digit Plus code and this guarantees centimeter precision. As soon as we move our avatar inside the monitor, the WDNS is updated to show only the live position. In Figure 7.10 we observe a scene with a box at the bottom left containing the geotag of the current WDNS. A feature has been implemented that allows you to scroll through the list of tags, using the Q and F keys on the keyboard, and to stay informed about the total number of geotags available. In Figure 7.10, for example, the geotag gives us information relating to the tag that a user left on the door object ("Door") also containing a review on the tagged object (bottom right box). Figure 7.11 represents a scene containing a box in the upper right corner showing a geotag of a different type containing information derived from the BIM model. An example is shown on a geotag with geometric information on the width of the door equal to 80 cm.



1/2 - Premere i tasti Q ed E per scorrere

Porta

Porta di ingresso ampia

WDNS
8F9QF3GM+P5MJM^21,2

Figure 7.10: Geotag Demo.

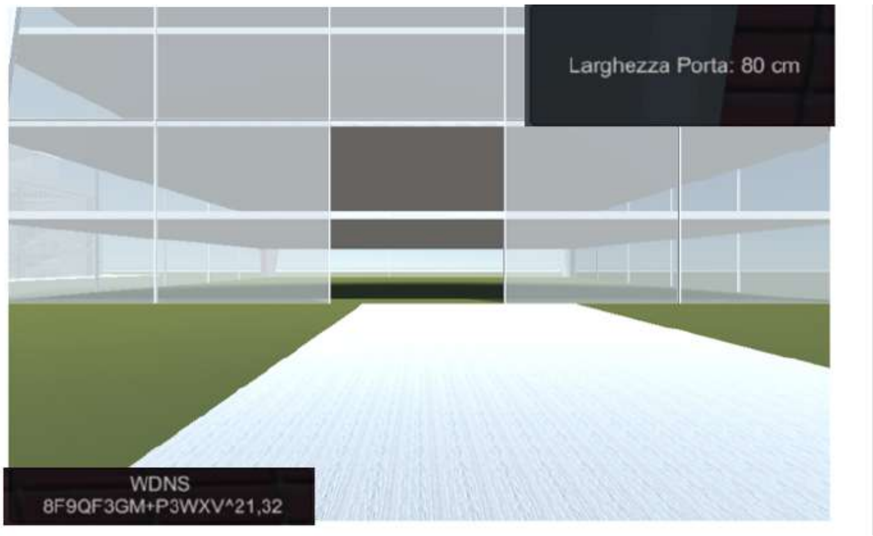


Figure 7.11: BIM Geotag Demo.

Although we have an approximate shape of the project we intend to create, we already have all the macro elements we want to insert: an accurate geolocation of the building, a good geocoding of the position in real time, a navigation algorithm to move the avatar within the virtual building and integration with the data layer that allows you to retrieve and place geotags from users and the BIM model.

7.2.6 Final Remarks

With this work we want to present an innovative prototype platform based on the concepts of Building Digital Twin and high precision Indoor Virtual Navigation, integrated with social functions based on posting and geotagging. With this platform we want to give users the opportunity to share an opinion, a review and a post by tagging a real place. Users who browse the same virtually reconstructed place will be able to read and interact with the posts associated with the different tags directly from home. In this way, a system of "trust" is created within the geolocalization functions of posts present in current social networks. In particular, we have tried to extend the complex concept of geotags, understood as location-based social network posts (especially on Twitter), by integrating WDNS, a three-dimensional geocoding algorithm capable of identifying any area of the world with the desired precision. As future works we want to apply our platform to Augmented Reality use cases, after applying indoor navigation as a path-finding tool inside a building. We also think that the implemented prototype platform can be used in more vertical use cases, for example for tourism, or for the identification of architectural barriers.

7.3 Scenarios for the Optimization of Citizen Movements

In recent years, governments have shown the need to create sustainable and technological advanced health systems [207]. Telemedicine is one of the major application domains that can positively impact people's lives. In this context, the concept of tele-biomedical laboratory is becoming, even more, a topic of interest among both the medical operators and the scientific community. It is a medical laboratory where blood exams are performed either by patients themselves in their homes or by biomedical technicians in satellite clinical centres through the Internet of Things (IoT) biomedical devices interconnected with Hospital Edge/Cloud systems that allow results to be manually or automatically sent to doctors of federated hospitals for validation and/or consultation. Biomedical laboratory health technicians, nurses, doctors and other clinical personnel belonging to different Federated Hospital IoT Clouds (FHC) cooperate to form a Virtual Healthcare Team (VHT) able to carry out a healthcare workflow. Nowadays, tele-biomedical laboratories are at an early stage. Currently, biomedical technicians are used to processing blood samples invasively taken from patients by nurses through syringes. The blood samples are typically processed manually through biomedical diagnostic devices. Often such diagnostic devices are not network-attached or vendor lock-in that depends on proprietary software with limited features and without open Application Program Interfaces (APIs). For economic reasons, clinical centres are reluctant to update their devices and manufacturers are not inclined to open APIs. These factors slow down the technology advancement toward tele-biomedical laboratory taking the advantages of the Cloud, Edge and Internet of Things (IoT) technologies. Furthermore, recently the scientific community is encouraging the development of new innovative minimally invasive or non-invasive biomedical diagnostic devices that exploits both the infra-red and Machine Learning (ML) technologies. This work aims at providing a clear picture about the state of the art towards near future tele-biomedical laboratories, highlighting where we currently are, issues and future challenges.

7.3.1 Motivation

Tele-biomedical laboratory allows performing patients' acceptance, blood exams, and results validation, in satellite clinical centres. This is possible utilizing the creation of a VHT composed of nurses, biomedical laboratory technicians and doctors belonging to different federated hospitals. Cooperation is possible through FHC that can involve several satellite clinical centres belonging either to the same healthcare organization or to different ones.

An example of a healthcare organization including different hospitals is the provincial healthcare organization of Messina (Italy), also referred as ASP Messina. As shown in Figure 7.12 it includes eight healthcare districts including, Messina, Taormina, Milazzo, Lipari, Barcellona Pozzo di Gotto, Mistretta and Sant'Agata di Militello. The healthcare district of Lipari is placed on the island of Lipari and provides a limited number of health services. It offers first aid to patients using an emergency room and a biomedical laboratory of clinical pathology. Due to the limited number of health departments, patients with particular diseases are typically transferred to the near healthcare districts of Milazzo or Barcellona Pozzo di Gotto (indeed by helicopter for urgent cases) if required. In this scenario, a tele-biomedical laboratory service could help the accomplishment of a clinical workflow involving a VHT including nurses, biomedical technicians and doctors belonging, for example to the Lipari, Milazzo and Barcellona Pozzo di Gotto districts. In particular, blood exams could be performed in the medical laboratory of Lipari by biomedical laboratory technicians and results can be transmitted using the FHC environment to a doctor of the Barcellona Pozzo di Gotto district for validations. Furthermore, leveraging the FHC environment an additional consultation could be done with a doctor of the Milazzo district.

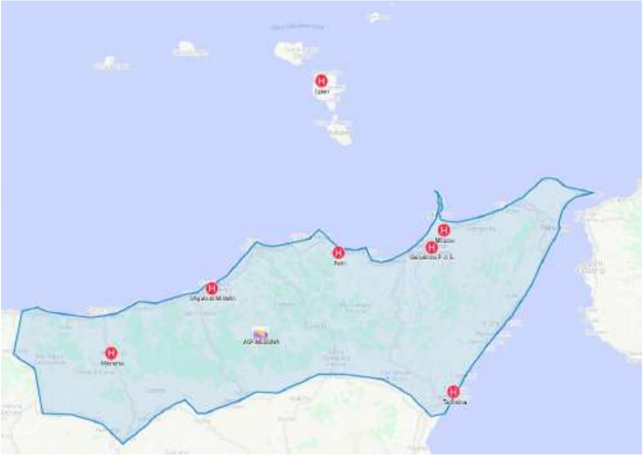


Figure 7.12: ASP of Messina: an example of healthcare organization spread over different healthcare districts.

Defining with the term "home hospital" the hospital that is physically reached by the patient, the generic healthcare workflow accomplishing the aforementioned scenario implies the following phases:

1. **Hospitalization**: patient reaches a home hospital; personal details, date and type of visit are recorded; the patient is identified by a visit identification code; an operators schedule blood exams. A VHT is created involving nurses, biomedical technicians, and doctors belonging to the home hospital and other federated hospitals;

2. **Clinical Analysis**:a nurse of the home hospital takes a blood sample from the patient; a biomedical laboratory technician of the home hospital performs blood tests through biomedical equipment and results are either automatically or manually sent to the FHC;
3. **Validation**: a doctor belonging to the FHC analyzes and validates the received results of clinical analysis.

An alternative challenging, tele-biomedical laboratory scenario, could be accomplished considering patients who perform themselves blood exams in their own homes through innovative remote IoT biomedical devices and transmitting results either automatically or manually with the human intervention to the FHC. The use of these innovative techniques allows us to limit the travel of specialists and patients. This is certainly useful in case of emergency, but it can help reduce user travel with all the benefits it brings.

7.3.2 Recent Advancements Biomedical Laboratory Diagnostic Devices

Tele-biomedical laboratory ideally requires connected IoT biomedical diagnostic devices able to assess human blood samples and to automatically send results to the FHC. However, most biomedical devices currently adopted in clinical centres are disconnected and consequently, they require the intervention of clinical operators to manually send results to the FHC. Furthermore, different tele-biomedical laboratory scenarios are possible according to the adoption of invasive, minimally invasive and non-invasive biomedical diagnostic devices able to perform blood exams. In this Section, we provide a discussion about such devices useful to present different tele-biomedical laboratory scenarios from the simplest one to the more complex, challenging and futuristic one.

Connected and Disconnected Devices

A biomedical laboratory device is a piece of equipment able to perform several blood exams including CBC, Basic Metabolic Panel, Complete Metabolic Panel, Lipid Panel, Thyroid Panel, Enzyme Markers, Sexually Transmitted Disease Tests, Coagulation Panel and DHEA-Sulfate Serum Test. Currently, there are many medical laboratory devices available on the market. A classification can be done considering "connected and "not connected" devices. For "connected" devices we intend biomedical laboratory diagnostic equipment including USB and network (wired and/or wireless) interfaces and able to export and send results to other devices, whereas for "not connected" devices we intend biomedical laboratory diagnostic devices without any interface for data transmission. In the following, we provide an overview of the major "connected" diagnostic devices that are based on future tele-biomedical laboratory services.

Telemedicine Clinical Monitoring Unit (CMU) [208] is a medical device able to perform blood pressure, pulse oximetry and blood glucose exams. Enverse [209] is a device able to perform continuous glucose monitoring. It consists of a chip that is installed subcutaneously on the patient that is connected with a mobile app. Med-Care [210] is an integrated solution for the auto-monitoring of glycemia that works with both web and mobile systems and that can send alerts via email or SMS. HemoScreen [211] is a low-cost portable haematology analyzer that performs a complete blood count at the point of care including a local web interface. Samsung Labgeo PT10S [212] is a portable clinical chemistry analyzer that improves efficiency by saving time for clinicians and patients through fast, easy and accurate blood analysis. It includes an ethernet interface to export exam results in an external Personal Computer (PC).

Invasive, Minimally Invasive and Non-Invasive Techniques

Currently, apart from common invasive biomedical diagnostic devices for the assessment of human blood levels, alternative emerging minimal invasive and non-invasive diagnostic devices are the argument of study for both academic and industrial healthcare communities.

Invasive Human Blood Tests

The invasive approach for blood exams is the most commonly adopted practice in clinical centres. Although it does not represent a risk for the patient's health, it presents management costs that can not be negligible. Such costs included administrative personnel for hospital acceptance, nurse performance to collect the blood sample from the patient, biomedical technician performance for blood tests assessment, use of biomedical diagnostic devices and reagents, and medical personnel performance for validation.

Minimally-Invasive Human Blood Tests

In the minimally invasive approach, only a few drops of blood are required to calculate exam results. Specifically, blood images and spectra-based information are captured from the blood sample for estimation. Typically, it considers a smartphone application as a point-of-care tool.

Non-Invasive Human Blood Tests

In the non-invasive approach, the data acquisition about the blood sample is performed through spectroscopic information acquire by a spectroscopy sensor. The acquired row signal is passed to a data pre-processing component that extracts features. In the end, the pre-processed data are sent to a database for further processing by means of ML algorithm, estimation and validation [213]. As well as the minimally invasive approach, even the non-invasive one typically considers a smartphone application as a point-of-care tool.

About the development of non-invasive haemoglobin level assessment solutions using a smartphone as a point-of-care tool, recommended data collection techniques, signal extraction processes, feature calculation strategies, theoretical foundations, and ML algorithms are discussed in [213]. An approach for the haemoglobin level assessment using a photoplethysmographic (PPG) sensor is discussed in [213]. An alternative approach has been implemented by HemaApp, a mobile application able to perform non-invasive haemoglobin measurement using unmodified smartphone cameras and built-in LEDs [214].

A non-invasive optical plethysmographic measurement of blood hematocrit approach was developed in the "Non-invasive measurement of blood water content using infrared light" project developed by Amsterdam UMC, BME and Physics, and Haemo Pulse [215]

A non-Invasive blood glucose measurement approach using a near infra-red spectroscopy sensor integrated into a diagnostic device that has to be worn on a fingertip is discussed in [216]. Specifically, a quantitative correlation between the concentration of glucose and of near infra-red radiation is established. A similar device using a ML strategy to assess the glucose level is discussed in [217]. The device's sensor consists of a pair of LED and photodiode which transmit and receive light with a wavelength of 940 nm. The light intensity reading from the sensor is amplified and filtered to reduce noise, then transmitted to the smartphone. In the smartphone application, the reading results are converted to blood glucose level using a ML model embedded in the application itself. An approach assessing the human blood component levels from fingertip video Using DNN Based Models is presented in [218]. Another approach for the evaluation of a near-infrared light ultrasound system as a non-invasive blood glucose monitoring device is discussed in [219].

A continuous non-invasive blood pressure approach for the measurements in humans under hyperbaric and/or oxygen-enriched conditions is discussed in [220]. Specifically, acquired measurements were matched with intermittent ones related to brachial arterial pressure. An alternative ML-based approach for non-invasive blood pressure estimation is discussed in [221].

In this context, recently person-specific blood-based infrared molecular fingerprints are opening up interesting perspectives for patient's health monitoring [222].

7.3.3 Tele-Biomedical Laboratories Scenarios

Tele-biomedical laboratory aims at improving the efficiency of the whole clinical workflow, allowing patients to better self-care themselves also improving their quality of life, and reducing clinical costs. Considering the classification of biomedical diagnostic devices discussed previously, in this Section, we present several possible future tele-biomedical laboratory scenarios that allow sending patient's blood exam results to a VHT through an advanced Cloud/Edge infrastructure interconnected with next-generation network-attached invasive, minimally invasive and non-invasive biomedical IoT diagnostic devices.

Figure 7.13 shows a high-level tele-biomedical laboratory architecture. Diagnostic devices are connected to a Local Edge Unit (located directly in the patient's home or in a decentralized satellite hospital centre) which will send the data, via a secure channel through the Internet, to a Hospital Cloud Unit. Blood exams can be performed through either invasive, minimally invasive or non-invasive biomedical diagnostic devices. Furthermore, in the case of connected devices results are automatically sent to the Hospital Cloud Unit, whereas in the case of non-connected biomedical devices it is required the human intervention either of the patient or the clinical operator who fill in a web form to send the blood exam results to the Hospital Cloud

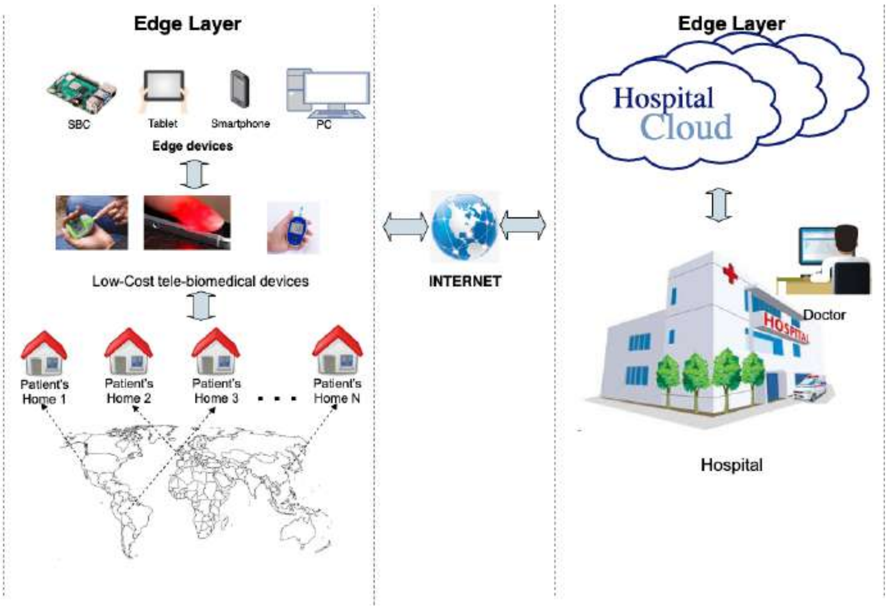


Figure 7.13: Tele-biomedical laboratory architecture.

Unit. Local processing can be performed with the support of Edge devices. The Hospital Cloud Unit receives remote blood exam results and it stores and processes them to provide different service levels, that are, Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Function as a Service (FaaS) and Software as a Service (SaaS) to develop different Hospital information system tools. Furthermore, different Hospital Cloud Units can cooperate in the FHC environment to support VHT.

7.3.4 Final Remarks

Tele-biomedical laboratory is at an early stage. The major issues that in recent years have slowed down its development have included:

1. the lack of non-invasive biomedical devices;
2. the consequent need of a clinical operator in presence to collect the patient's blood sample;
3. the fact, that the market of biomedical devices has been mainly dominated by vendor lock-in solutions.
4. the reluctance of manufacturers to include in their biomedical devices open APIs;
5. the concerns of hospitals about data security;
6. the reluctance of hospitals to port their systems over the Cloud due to economical reasons;
7. the poor maturity of the Cloud, Edge and IoT technologies;
8. the reluctance of clinical operators to revolutionize their habits and healthcare workflows.

Nowadays, we believe that the Cloud, Edge and IoT technologies are mature enough and that Hospitals begin to open toward telemedicine services also considering the recent large capital investments promoted by governments for the digitalization of the healthcare sector [207]. Also patients' security and privacy do not represent a big concern anymore due to the recent technological advancements. Also, the costs reduction of even more connected biomedical devices is pushing the rising of tele-biomedical laboratory services. Of course, the current biggest challenges regards the development and adoption on a large scale of even more sophisticated non-invasive biomedical diagnostic devices based on both the infra-red and ML technologies. The latter is still at an early stage, even though more and more scientific works are appearing in the literature on this topic.

For these reasons, we believe that the time is mature for the development of tele-biomedical scenarios consisting of remote Local Edge Units using invasive and minimally invasive IoT biomedical devices placed in satellite clinical districts interconnected with Hospital Cloud Units. Also, the FHC ecosystem is technically possible even though there are still several bureaucratic barriers to be felled. However, we believe that there is still a long way to go towards the development of Local Edge Units placed in the patients' homes adopting non-invasive IoT biomedical diagnostic devices even though recent related works have focused on the assessment of haemoglobin, hematocrit, glucose, and blood pressure levels.

With this work, we hope we succeeded in stimulating the interest of both scientific and industrial communities about the development of the near future tele-biomedical laboratory even for reduce the movement of people.

CHAPTER 8

Custom Service in Edge/IoT Devices

Providing services in smart environments means physically deploying IoT or Edge devices. The devices are necessary both for data collection through sensors and for the implementation of automatic operations. These needs have increased the number of devices in the environment and this has a negative impact in terms of system scalability and sustainability. As a solution to this problem, we propose to customize the services running on Edge and/or IoT devices so that a single device can be used to the limit of its capabilities. The device will thus be able to offer different types of services that will be configured on-demand. How possible solution we introduce the concept of the "virtual device", which is an abstracted component characterized by specific high-level functionalities. This chapter proposes a data model useful to represent and optimize the adoption of "virtual devices" in smart environments. A definition of ”virtual sensors” was proposed, which are abstracted components able to map different behaviors on the same Internet of Things (IoT)-based infrastructures according to the needs of the high-level applications. To realize ”virtual sensors”, it is necessary to codify user requests in an automation process for the deployment at the Edge of the microservices (MSs) that satisfy such requests. We present a solution that implements all the necessary functionalities to bind the user application with the Edge device in charge of executing the "virtual sensors”. Another aspect reported in this chapter concerns the way services are distributed in smart environments, approaching the computation where data are generated. In these cases, Edge computing represents a challenging solution supporting IoT with flexible management of resources. During our study, we investigated how pushing computation activities from Edge to IoT, changes the behavior of IoT nodes according to application or system requirements. We adopted the Multi-Hop-Over-The-Air update technology enabling the auto-configuration of IoT devices based on MicroController Units. Considering IoT nodes connected in a mesh network, we developed a distributed and collaborative ecosystem performing on-fly injection of code in IoT nodes, thus automatically deploying new services whenever necessary. The experimental results of the study are reported in this chapter.

8.1 Protocols for Optimizing Edge Devices

Smart environments are challenging contexts to offer a new generation of services on the basis of huge amount of gathered data and pervasive processing. In particular, distributed Edge computing solutions move data processing closer to end-users, handling data in real time, providing local feedback and adding robustness to connectivity. Many technologies can be exploited in smart environments for implementing Edge computing solutions, interconnecting different hardware and software components and deploying services and applications. This high heterogeneity of available resources need to be represented in a abstracted way, in order to provide a useful solution for the high level management of both simple and complex smart environments. One of the key challenges of Edge computing concerns the exchange of data between the various actors of the smart ecosystem. In this field, FIWARE NGSI [223] provides a standardized approach for the exchange of context information. NGSI-LD is an information model that allows applications to perform dynamic and flexible discovery and query of data, also getting information on the related context, such as the period of validity, geographic constraints, and other semantically important information. Even if NGSI-LD is valuable for retrieval and interpret data, it cannot represent how, where and when data has to be manged and processed. Today we identify a limit in the existing data models, that characterize only the data and the data source, but not the services that allow data to be acquired or managed. Describing smart systems and possible changes in their behaviour is a big challenge that could simplify the deployment of services and applications in the same environment. In this paper, we introduce the innovative concept of Virtual Device, an abstracted component able to describe the behaviour of a computing node from the point of view of its potential in processing functionalities. To deal with the deployment of Virtual Device, an extension of the NGSI-LD standard is proposed in order to enrich data with related processing information. Our approach makes it possible to create geolocalized data models and provides designers and developers high flexibility in the characterization of a service deployment system based on the FaaS technologiy [224]. We show how it is possible to deploy multiple Virtual Devices on the same physical device simply by configuring the related data model. Finally, the proposed approach is applied to a specific use case by introducing the concept of solid angle in the video surveillance applications of a Smart City.

8.1.1 Related Work

An interesting survey on Edge computing opportunities for smart cities has been carried out in [225], where authors highlight the role of Edge computing to realize the vision of smart cities with the objective to classify the literature by devising a comprehensive and meticulous taxonomy. They identify and discuss key requirements, and enumerate recently reported synergies of edge computing enabled smart cities. Finally, several indispensable open challenges along with their causes and guidelines are discussed, serving as future research directions. Intelligent offloading for collaborative smart services in Edge computing [226] claims the weakness of long service response time and low QoS in scenarios with clouds. The authors remark that edge computing is nowadays integrated with the smart city to promote the inherent shortcomings of terminals in cities, to this they designed an intelligent offloading method for collaborative smart city services, named IOM. They try to achieve a trade-off among minimizing service response time, optimizing energy consumption and maintaining load balance while guaranteeing the privacy preservation during service offloading. A comprehensive and good analysis with mathematical models has been done. Given the ever-increasing need to send and receive information from collection devices (sensors, cameras, etc.) to cloud computing devices [227][228], the risk of increasing the percentage of errors during data transmission and packet loss is inevitable, especially in cloud and edge-based network architectures. Studies [228] and [227] aim at reducing the computational load of cloud devices to redistribute it to other hardware, according to performance improvement logics, and migrating the same load to computationally less powerful but very useful devices, positioned as interface infrastructures between field and cloud devices, the so-called edge devices. The possible computations in these devices concern a data pretreatment [229][228] consisting of an encapsulation of the information within a pre-packaged data-model that allows a better management and analysis, a "filtering" to skim the number of data to travel on the network and reduce the number of consignments, which could lead to an improvement in terms of reduction of transmission errors and longer battery life in the case of devices not powered by current (energy consumption). With regard to the computational capabilities of edge devices, [230] highlights the possibility of obtaining, in addition to the "raw" data, also "indirect" information extrapolated from the individual measurements (in the case of sensors) fed into special statistical calculations or "predictions/predictions" made on dynamic information. [231] instead highlights the great capacity of job sorting within a network made up of cloud and edge devices, which can be managed and reconfigured through JSON messages and with a micro-service oriented architecture. In [232] it is highlighted how necessary a data model that offers advanced functionality for the description of the context can exploit different data sources. However, the data can be used in various contexts for different solutions. The authors highlight how FIWARE technologies help the scientific community and developers through the data models defined by ETSI ISG Context Information Management (ETSI CIM). The work concludes by highlighting how the development of IoT technologies is driven by semantic and context-sensitive data models. From the considerations carried out it emerges that interoperability and models with semantic annotations significantly increase the reusability of IoT resources outside their initial specificity. The growing use of IoT device applications in smart environments, associated with a greater demand for computational challenges, today sees cloud infrastructures increasingly used and with the need to rationalize resources. Regarding this new need, in [148] it is highlighted how it is possible to use customized generic Edge devices to carry out multiple activities simultaneously can be a solution to lighten the work of cloud infrastructures. The authors have implemented and tested, in a real solution in the city of Messina (Italy), a solution based on the Function as a Service (FaaS) paradigm. The proposed solution allows users to perform multiple activities on the same device such as vehicle counting, license plate recognition, identification objects etc. In this case, two cameras were connected to a Raspberry PI 4 and the performance compared. Nothing prevents you from connecting different sensors to the Edge device and imagining each sensor as a different service. Each service can be managed through the concept of Virtual Device with the use of specific Data Model. Furthermore, it is possible to imagine not only a single Virtual Device, but also a Virtual Device network [233]. A Virtual Device network is composed by services deployed on different Edge devices. Each device in the network with a specific service can work with the others in order to perform complex processing for services in a given area. In this section we want to put together the concepts analyzed so far in order to formulate a data-model that, through the architectural paradigm of micro-services, allows to simply manage the configuration and display of field devices, the reduction of the amount of data transmitted, energy consumed and transmission errors, offering more services resulting from the installation of a single device (sensor, camera or IoT), a concept that we will present under the name of virtual-device.

8.1.2 What is NGSI-LD

NGSI-LD is an open standard for managing context information. NGSI-LD was released in 2018 as an ETSI specification to enhance FIWARE's NGSIv2 standard. The improvement that led to the development of NGSI-LD concerns the improved support of linked data and the definition of properties and semantics structured according to the JSON-LD standard. NGSI-LD is based on the key concepts of: Entity, Property and Relationship. Entities are objects with specific properties and can be in relationship with other Entities. Properties are a combinations of attributes in the form $(key : value)$. Relationships allow to establish logic connection instances through linked data using a property that points to another external resource (identified by a Uniform Resource Identifier (URI)). This characteristic is derived from JSON-LD and allows to define and connect entities in a clear and unique way. Properties and relationships can be part of other properties or relationships along a maximum of 1 or 2 levels of concatenation in NGSI-LD schemas. Properties characterize the entities involved and their interpretation is defined in the *@context* component. This feature allows the generalization of the data model and marks the difference between the NGSIv2 and NGSI-LD formats. Only some properties are standardized so to be implicitly defined and not contained in *@context*. To better explain the key features of NGSI-LD, in Listing 8.1, we show a well-known example of a NGSI-LD data model for a sensing device [234].

```
{  "id":"urn:ngsi-ld:Device:device-9845A",
    "type": "Device",
    "category": {
        "type": "Property",
        "value": ["sensor"]
    },
    "batteryLevel": {
        "type": "Property",
        "value": 0.75
    },
    "dateFirstUsed": {
        "type": "Property",
        "value": {
            "@type": "DateTime",
            "@value": "2014-09-11T11:00:00Z"
        }
    },
    "controlledAsset": {
        "type": "Relationship",
        "object": ["urn:ngsi-ld::wastecontainer-Osuna-100"]
    },
    "value": {
        "type": "Property",
        "value": "l%3D0.22%3Bt%3D21.2"
    },
    "refDeviceModel": {
        "type": "Relationship",
        "object": "urn:ngsi-ld:DeviceModel:myDevice-wastecontainer-sensor-345"
    },
    "rssi": {
        "type": "Property",
        "value": 0.86
    },
    "controlledProperty": {
        "type": "Property",
        "value": ["fillingLevel", "temperature"]
    },
    "owner": {
        "type": "Property",
        "value": ["http://person.org/leon"]
    },
    "deviceState": {
        "type": "Property",
        "value": "ok"
    },
    "@context": [
          "https://schema.lab.fiware.org/ld/context",
          "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.
    jsonld"
    ]
}
```

Listing 8.1: Example of NGSI-LD data format

The *id* attribute is an Uniform Resource Name (URN) that can be used by other entities to link this one; the attribute *type* clarifies what kind of entity it is. The *category* attribute specifies the assets of the attribute type. *ControlledAsset* and *refDeviceModel* link the entity with other entities, whereas the remaining attributes represent the current values (e.g.,)payload) in the data model. Their structure, if not standardized, is defined in the *@context* attribute.

8.1.3 The NGSI-LD Extenxion for Representing a Virtual Device Model

A Virtual Device is an abstraction of a physical device that emphasize specific features or functionalities of the device itself.

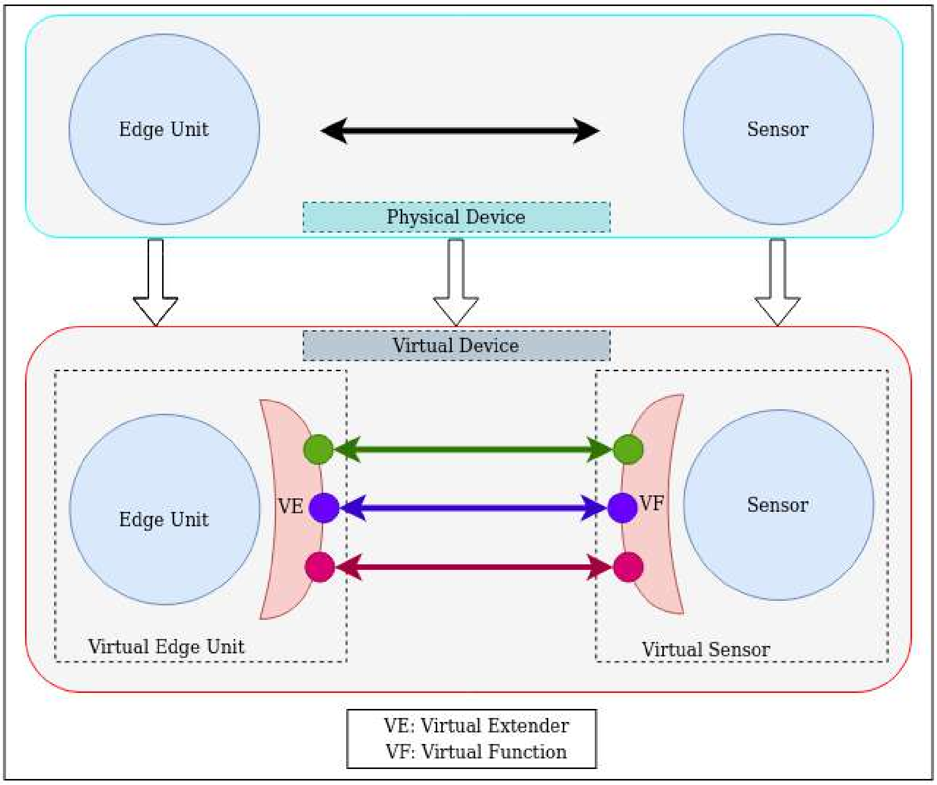


Figure 8.1: Abstracting schema of a Virtual Device.

This concept is outlined in the Figure 8.1: the Edge Unit is a physical device at the edge and it can be connected to a sensor (or more sensors) for gathering (and after processing) data from the environment, such as recording the measurement of temperature at specific time intervals. With this approach, data collection and processing tasks are strictly related to the type of binding between the Edge Unit and the sensor. A Virtual Device maps several behaviours of the Virtual Edge Unit with the ones of the Virtual Sensor. As shown in the lowest part of Figure 8.1, the Virtual Device allows to decouple different possible activities executed into the Edge Unit by using data coming from the sensor, such as estimating the average temperature, identifying peaks in temperature measurements, and so on, thus increasing the flexibility in data processing. In particular, through a Virtual Extender (VE), it is possible to connect N-functions (coloured dots in Figure 8.1) in the Virtual Edge Unit with the same amount of functions in the Virtual Sensor. In this way, it is like N-different physical devices are available. The VE together with the Edge Unit identifies the Virtual Edge Unit, which communicates (see the coloured arrows) with the Virtual Sensor by means of the Virtual Function (VF) component. The VF together with the Physical Sensor define the concept of Virtual Sensor. The Virtual Edge Unit and the Virtual Sensor compose the Virtual Device.

Considering the capabilities reached by current Edge computing devices, it is possible to image several independent processes running at the Edge that elaborate data in a different ways. This entails a considerable economic advantage in the deployment of Edge solutions on a large scale, but also a considerable advantage in terms of environmental protection. In Figure 8.2 a schematic example is shown.

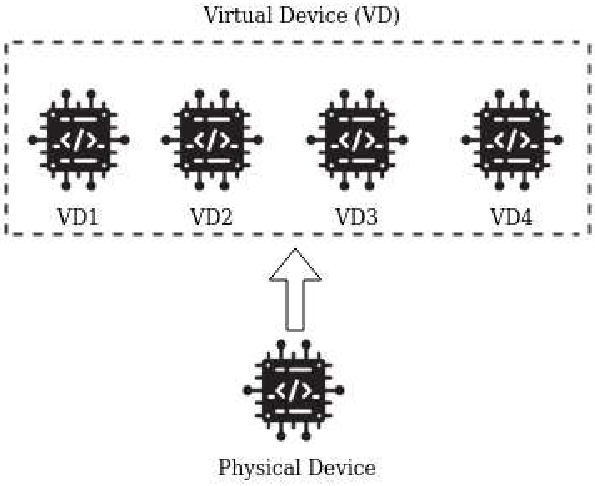


Figure 8.2: General representation of a Virtual Device.

The physical device can be, for example, a micro-controller with a temperature sensor attached. The Virtual Device VD1 can provide data in real time, the Virtual Device VD2 the daily average temperature, the Virtual Device VD3 the weekly average temperature and the Virtual Device VD4 the maximum weekly temperature value. The limit of the number of virtual devices depends only by the processing limits of the micro-controller. Applications asking for the service will see a dedicated service, but, in reality, they will access information processed at the Edge. The request and the access to the service from an application need to be defined and managed through a data model. To include the concept of Virtual Device within the NGSI-LD data model, it is possible to create an extension of the standard that abstracts the concept of physical device. The extension is accomplished by adding information to the payload of the model, thus to have a more flexible data model.

```
{
    "id": "urn:ngsi-ld:Device:device-9845A",
    "type": "Device",
    "category": {
        "type": "Property",
        "value": ["sensor"]
    },
    "batteryLevel": {
        "type": "Property",
        "value": 0.75
    },
    "deploymentInfo":{
        "version":1.0,
        "provider":{
            "name": "openfaas",
            "gateway": http://127.0.0.1:8222
        }
        "type": "functions",
        "datafunctions-arm":{
            "lang": "python3-flask-debian",
            "handler": "./datafunctions-arm",
            "image": "urbanite-messina/datafunctions-arm:latest",
            "environment":{
                "physical_dimension": "meteo",
                "measurement": "Temp",
                "operation": "deploy",
                "date": "2018-04-06T11:00:00Z",
                "place": "S.Marco",
                "read_timeout": "50s",
                "write_timeout": "50s",
                "upstream_timeout": "50s",
                "exec_timeout": "50s"
            }
        }
    },
    "dateFirstUsed": {
        "type": "Property",
        "value": {
            "@type": "DateTime",
            "@value": "2014-09-11T11:00:00Z"
        }
    },
    "value": {
        "type": "Property",
        "value": "22.3"
    },
    "refDeviceModel": {
        "type": "Relationship",
        "object": "urn:ngsi-ld:DeviceModel:urbanite-device-sensor-temp"
    },
    "rssi": {
        "type": "Property",
        "value": 0.86
    },
    "controlledProperty": {
        "type": "Property",
        "value": ["fillingLevel", "temperature"]
    },
    "owner": {
        "type": "Property",
        "value": ["http://fcrlab.unime.it"]
    },
    "deviceState": {
        "type": "Property",
        "value": "ok"
    },
   "location": {
        "type": "GeoProperty",
        "value": {
            "type": "Point",
            "coordinates": [38.1885046,15.5535013]
    },
    "@context": [
        "https://schema.lab.fiware.org/ld/context",
        "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.
   jsonld",
        "fake:https://fcrlab.unime.it/urbanite-messina"
    ]
}
```

Listing 8.2: Extension of NGSI-LD Physical Device data format

In Listing 8.2, an example of the extension of NGSI-LD is shown in the case of an Edge device with the ability to support Virtual Device. The device is linked to a specific model by the *refDeviceModel* property, but thanks to the *deploymentInfo* property it can be set to host Virtual Device. The *deploymentInfo* property is read and converted into a file yaml which is used to call the openFaas function which deploys or starts the service as in this case. Each field of the *deploymentInfo* attribute will then be defined in *@context*. The location attribute determines the location of the physical device.

```
{
    "id": "urn:ngsi-ld:Device:device-9845A-Pugliatti-1",
    "type": "VirtualDevice",
    "category": {
        "type": "Property",
        "value": ["Virtualsensor"]
    },
    "deploymentInfo":{
        "version":1.0,
        "provider":{
            "name": "openfaas",
            "gateway": http://127.0.0.1:8222
        }
        "type": "functions",
        "datafunctions-arm":{
            "lang": "python3-flask-debian",
            "handler": "./datafunctions-arm",
            "image": "urbanite-messina/datafunctions-arm:latest",
            "environment":{
                "physical_dimension": "meteo",
                "measurement": "Temp",
                "operation": "average",
                "date": "2018-05-06T11:00:00Z",
                "place": "S.Marco",
                "read_timeout": "50s",
                "write_timeout": "50s",
                "upstream_timeout": "50s",
                "exec_timeout": "50s"
            }
        }
    }
    "value": {
        "type": "Property",
        "value": "21.1"
    },
    "refDevice": {
        "type": "Relationship",
            "object": "urn:ngsi-ld:Device:device-9845A"
        },
        "controlledProperty": {
            "type": "Property",
            "value": ["DailyAveragetemperature"]
        },
        "deviceState": {
            "type": "Property",
            "value": "ok"
        },
        "@context": [
            "https://schema.lab.fiware.org/ld/context",
            "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.
    jsonld",
            "fake:https://fcrlab.unime.it/urbanite-messina"
        ]
}
```

Listing 8.3: Extension of NGSI-LD Virtual Device data format

In Listing 8.3 an example of Virtual Device deployment is shown. In this case, respect to the Listing 8.2, it is sufficient to change the value of "operation" in the *infoDeployment* property to deploy a new service that returns the weekly average of the measured temperature. The virtual device is tied to the physical device by the *refDevice* property. By doing this, it is possible to deploy different services simply by changing a field or querying them and defining the various rules in the payload of the proposed NGSI-LD extension. The Virtual Device can be deployed or stopped as needed, but the physical device will always be ready to host new services. It is also possible to think of different types of sensors on each device. Each sensor can then be virtualized by creating heterogeneous Virtual Devices on the same physical device.

8.1.4 Use Case: Smart Environment Monitoring With a Virtual Camera

The use case we analyzed in this paper concerns a video surveillance system in a smart city. Edge devices are designed to offer miscellaneous micro-services; based on the actor specific measurement request in a precise moment a single device can offer one or more aggregated data obtained by different sensors integrated on the same board. The only limit for requests specifications is the physical one. To simplify the concept: an ip-cam can acquire 25 frames per second (physical limit for this device). The Virtual Device could offer to a specific actor 10 frames per second, but it will never can offer 30 frames per second (fps). In order to help this type of requests that can be differentiated but always addressed to the same device, we introduced the concept of Virtual Device as an abstraction of the physical device. In particular, in our vision, a Virtual Device is a extension of an NGSI-LD Physical Device definition type in which is introduced the specific configuration (e.g. 5 fps, 10 fps or 25 fps or solid angle). In order to further improve and enrich the data model that we want to propose, it has been thought to add a measure that can give an account of the angle and the surface that can be covered by the "framing" of a device that can acquire images from a given scene. This measurement has been identified with the ***solid angle***.

Before arriving at a more precise description of the solid angle, it is necessary to make a premise by starting to describe what is called *Field of View*. Considering Figure 8.3[1], he Field of View (**FoV**) is a measure of the observable world that can be seen at a given time. In the case of optical instruments or sensors it can be assimilated to the solid angle through which a detector is sensitive to magnetic radiation at a given time. Another well-known element in the field of photography is what is usually called the *Angle of View* (**AoV**) which describes the angular range of a given scene framed by a camera. The term angle of view can usually be assimilated with the more general term Field of View.

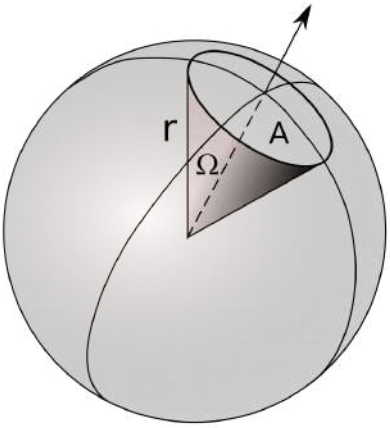


Figure 8.3: Solid Angle representation.

Returning to the solid angle, we can define it as the extension in three-dimensional space of the plane angle. The unit of measurement of the solid angle is the *Steradian* and can be

[1] Source: `http://pngegg.com`

calculated as:

$$\Omega = A/R^2(SolidAngle)$$

In which: **A** is the area of the spherical portion of radius **R** seen under the angle.

As we can see from the previous formula we can note how the ratio, even in the three dimensions, between the portion of circumference, the radius and the subtended angle is maintained. As in the planar angle, the solid angle is the ratio between the area of the spherical surface and the radius of the sphere considered.

We can further determine the relationship between the two corners:

$$d\Omega = (2\pi sen\Theta)d\Theta$$

where Ω is the solid angle and Θ is the plane angle.

To get a clearer picture of the formulas expressed, let's try to imagine a light bulb in the centre of a sphere. For the whole sphere the solid angle through which the light rays pass is valid:

$$\Omega = S/R^2 = (4\pi R^2)/R^2 = 4\pi$$

If, instead of considering the whole sphere, we consider the only part of the spherical surface crossed by the light rays, using the differentials we obtain:

$$d\Omega = dS/(R^2) = (R^2\theta d\theta d\phi)/R^2 = sin\theta d\theta d\phi$$

where θ is the ***colatitude*** (angle from the north pole) and ϕ is the ***longitude***.

This value can therefore be useful to understand how large the portion of surface that a given camera can be able to frame given a certain angle. The solid angle in this case is a specific service deployed as Virtual Device. The solid angle calculated by taking as reference the point where the device is located and can be useful in a video surveillance service (Figure 8.4 [2]).

Virtual Device enriched data model

Going forward to the concepts of Virtual Device, Physical Device and the data model that we want to propose in order to realize micro-services on the Edge, we want to start to introduce the above mentioned concept of solid angle.

[2] Vienna 2020 imagine of repertory published on `wikipedia.com`"

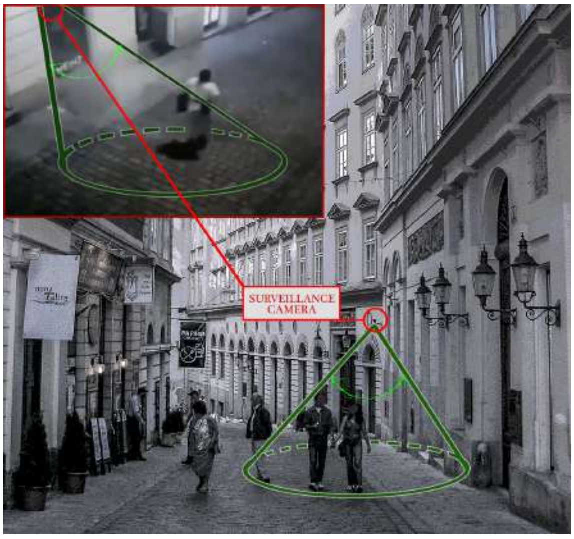


Figure 8.4: Solid Angle used in real scenario.

We can so describe the Virtual Device we want to define in our data model as shown in Listing 8.4.

```
{
    "id": "urn:ngsi-ld:Device:device-9845A-Pugliatti-2",
    "type": "VirtualDevice",
    "category": {
        "type": "Streaming",
        "value": ["5","10","25","solidAngle"]
    },
    "deploymentInfo":{
        "version":1.0,
        "provider":{
            "name": "openfaas",
            "gateway": http://127.0.0.1:8222
        }
        "type": "functions",
        "datafunctions-arm":{
            "lang": "python3-flask-debian",
            "handler": "./datafunctions-arm",
            "image": "urbanite-messina/datafunctions-arm:latest",
            "environment":{
                "operation": "solidAngle",
                "date": "2018-05-06T11:00:00Z",
                "place": "P.Pugliatti"
            }
        }
    },

    "refDevice": {
        "type": "Relationship",
        "object": "urn:ngsi-ld:Device:device-9845A"
    },
    "deviceState": {
        "type": "Property",
        "value": "ok"
    },
    "@context": [
        "https://schema.lab.fiware.org/ld/context",
        "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.
   jsonld",
        "fake:https://fcrlab.unime.it/urbanite-messina"
        ]
    }
}
```

Listing 8.4: NGSI-LD data format for a Virtual Device

The configuration, and the relative micro-service, can be changed and deployed in our Virtual Device, simply editing the "operation" field in Listing 8.4. Considering that an ip-cam could offer different micro-services and serve a multiplicity of users with different aims, i.e video quality, framing, frame rate, etc., the virtualization of the device, externally will give the impression that each user is talking to a different camera, but thanks to the processing on the Edge, the device will be physically one (Figure 8.5). Thanks to the data model designed the camera will be able to show itself virtually different to each user who makes a request.

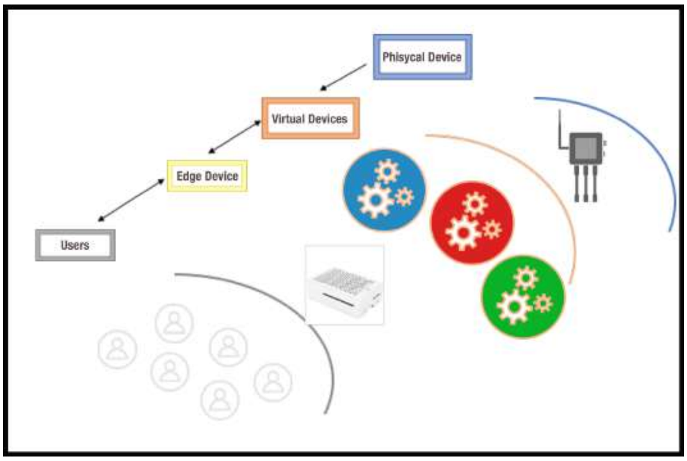


Figure 8.5: Virtual Device - high level schema

8.1.5 Final Remarks

This section proposes an innovative method of managing services and related data in a smart environment at the Edge. A study was conducted on the current data models used for exchanging data among heterogeneous systems and especially to Edge devices in smart environments. To the best of our knowledge, there are not models to clearly describe both data and services that need to be executed, we presented the concept of Virtual Device, which extends the concept of physical device in an innovative way. Then, we presented how a Virtual Device can be described with an extension of the NGSI-LD standard, in order to increase the fleaxibility of the data model and specifying important information on how data are processed and services are provides. In the proposed use case, the concept of solid angle was presented as possible service of a Virtual Device for a video surveillance service and represented through the proposed data model. In future works, we would like to investigate the performance of Edge devices in case of service deployed using the proposed NGSI-LD based description.

8.2 On-Demand and Automatic Deployment of Microservice

The growing use of Internet of Things (IoT) devices poses several challenges to the scalability of information systems and even to environmental sustainability. Furthermore, the widespread use of low-cost sensors implies great flexibility in IoT device configuration and arises new opportunities in the development of applications and services. To catch such opportunities, sensing and computation functionalities need to be coupled in Edge devices, which execute geo-localized and IoT-oriented composable microservices (MS). An example can be referred to video surveillance in smart cities: cameras installed in well-specified geographical areas capture the same scene from different perspectives so providing different pieces of information that, merged together, give a 3D value to the video surveillance application [11]. Another aspect that could be evaluated in the same application scenario is image reliability, which depends on the distance and angle of recording. The above example shows how the same sensing device (in this case cameras) can provide added values according to the type of data processing performed by the Edge devices. Describing smart systems and possible changes in their behaviour is a big challenge that could simplify the deployment of services and applications in the same environment.

The idea behind this work takes up this challenge. We start introducing the concept of "virtual sensors", abstracted components able to map different behaviours of the same physical infrastructures according to the needs of the high-level application. Then, we propose a new approach to automatize the deployment of customized MS in the Edge devices for responding to on-demand requests of end-users. In particular, the solution we proposed is based on the FIWARE NGSI-LD [223] information model, which provides a standardized approach for the exchange of context information. NGSI-LD is an information model that allows applications to perform dynamic and flexible discovery and query of data, also getting information on the related context, such as the period of validity, geographic constraints, and other semantically important pieces of information. Althought NGSI-LD is valuable for the retrieval and it interprets data, it can not represent how, where and when data has to be managed and processed. Nowadays, we identify a limit in the existing data models, that characterize only the data and the data source, but not the services that allow data to be acquired and managed. To deal with the deployment of a virtual device, an extension of the NGSI-LD standard is proposed in order to enrich data with related processing information.

NGSI-LD helps us to standardize the communication among the different entities involved in the process, as shown in Figure 11.1. A user asks for an IoT-based application via a web app in execution in the Cloud that exposes the service. In the Cloud, the request is formalized through an NGSI-LD document containing all the necessary pieces of information on the MS composing the application, the infrastructure where to deploy MS and all the configuration parameters. The NGSI-LD document is transmitted to the Edge device in charge to host one or more MS, which automatizes the procedures to set up and execute MS. The NGSI-LD document will provide the Edge device with the information necessary to deploy MS, thus customizing the specific behaviour of the Virtual Sensor (VS) [1]. The proposed solution makes it possible to reduce the amount of specialized physical devices installed in the smart environment because the processing is customized on demand, and this can increase the

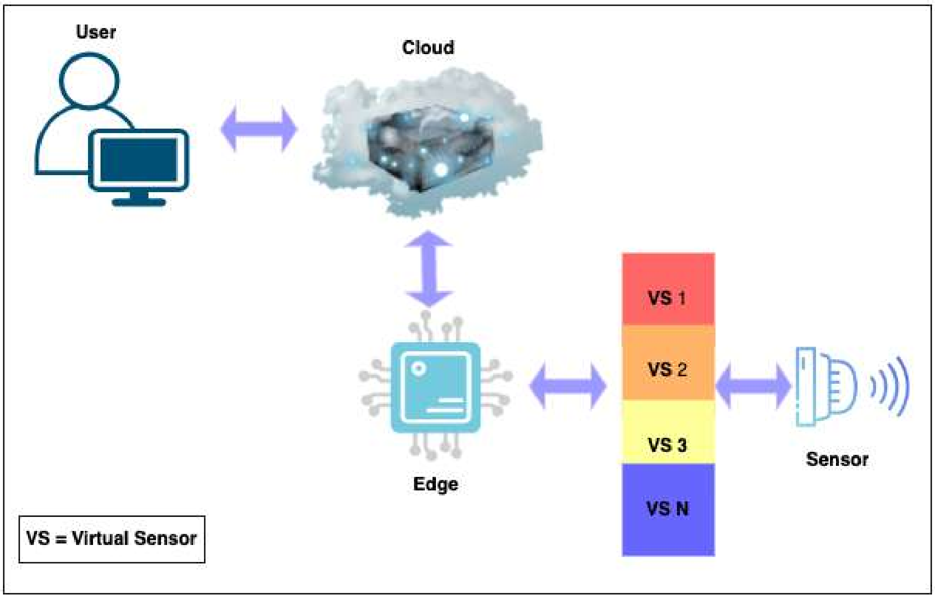


Figure 8.6: Scenario.

environmental sustainability of IoT-Edge installations.

8.2.1 State of the Art

The deployment of IoT devices has grown steadily in recent years. These equipments have different characteristics and levels of intelligence. The use of these devices often requires that they are able to self-test [12] [6] and configure themselves for specific tasks. In [235] an interoperability issue between Edge/IoT devices is addressed. In particular, the study highlights that standardized data models have been proposed to overcome the interoperability gap between different technological alternatives. Among the possible solutions proposed to improve the interoperability of the systems, we identify the data connected to the new interfaces of the generation service (NGSI-LD). The authors, starting from the consideration that the NGSI-LD information model is based on the JavaScript Object Notation (JSON) for Linked Data (JSON-LD) efficiently exploit the potential of semantics and linked data. In particular, it is pointed out that JSON-LD serialization is supported but not exploited due to difficulties in defining and implementing new Application Program Interfaces (APIs) based on NGSI-LD. Indeed, in the study, the authors propose an information mapping standard trying to demonstrate the simplicity of the standard with examples of real use cases and possible exploitation of semantic approaches. A similar study was also conducted in [1]. Here the authors describe how the NGSI-LD standard could be used in use cases that require dynamism. The study proposes descriptions of configuration documents to be used for the proposed use cases. The work proposed in this study starting from these considerations aims to extend the studies carried out by demonstrating in practice the benefits of this type of approach in the configuration of Edge devices in a smart city scenario. The described approach is also theoretically addressed in [236]. The authors propose a configuration based on JSON-LD to annotate semantically different types of services and semi-automatically allow their composition. The use case referred to in the study is a proof of concept. With our work, we want to demonstrate the possibility of applying this approach in reality by demonstrating its effectiveness. However, services in smart cities are increasingly being pushed from the remote Cloud to Edge sites close to data sources to offer fast response times and low power consumption. Configuration management tools that install software updates can also increase power demands, for example by stopping power management processes. In [237] the authors address these issues by proposing an automatic data-driven optimization approach. Our study proposes itself as a tool that facilitates communication between systems by supporting these types of policies. The purpose is to lighten the configuration/update policy of Edge systems. The problems described can be addressed by benefiting from containerization and MS technologies. This approach is used in [238]. In fact, the authors use the MS approach for on-fly configuration in the field of Fog computing. In their study, the authors demonstrate the goodness of their approach. In our work, we want to improve this approach by integrating the use of the NGSI-LD standard in the configuration of MS. The use of container-based technology has found wide use in the efficiency of monolithic architectures [239]. This technology is also finding wide use in Edge systems [240] and Edge-Cloud architectures as reported in [241] and [242]. In particular, in [242] the authors in their study are aware that research on container-based Edge computing is abundant due to the rapidity of deployment. Their study concerns the structures and algorithms on which planning models are built. In particular, the authors focus on the container placement problem which is mostly abstracted using multi-objective optimization models or graph network models. The study finds that there is a paucity of container scheduling models that take into account distributed Edge computing tasks. Our work aims at demonstrating that the approach used by the authors can be integrated and improved thanks to the use of the NGSI-LD standard for Cloud-Edge communication. The state of the art demonstrates how the research of methods of self-configuration and management of Edge systems is of fundamental importance for the development of complex ecosystems. The proposed work aims at demonstrating how the NGSI-LD model can be integrated into the deployment of services on Edge devices. Furthermore, with the work proposed in this paper, we want to define the basis for a concrete technological advance with respect to the state of the art.

8.2.2 NGSI-LD for describing MS deployment

Virtualization is an abstraction of a physical device that emphasizes specific features or functionalities of the device itself. In IoT-oriented applications, sensing devices have a key role because how data is collected and processed impacts the performance both the high-level application relying on that data. A "virtual sensor" is an abstraction of a physical sensor aimed at a specific acquisition and processing of sensed data. An Edge node supports data processing locally, which means close to the sensing area and, hence, it can be considered as an extension of one or more sensor devices.

The NGSI-LD document has to describe the infrastructural requirements for the deployment of MS. It has the characteristic of being customizable, for this reason, a functional structure has been used for the purpose of our work. A graphical representation of an example NGSI-LD document is shown in Figure 8.7.

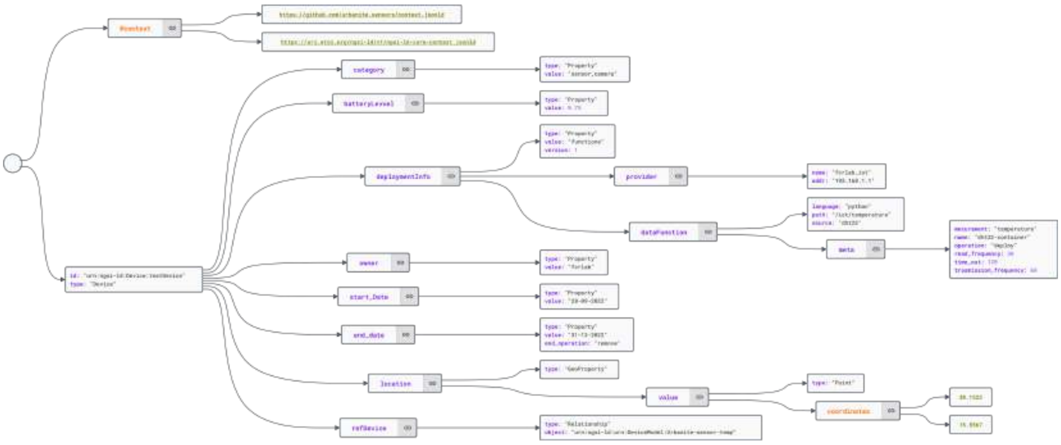


Figure 8.7: A graphical representation of an NGSI-LD document example.

The NGSI-LD document describes the device entity, identified with an ID. This is a service to be deployed that represents the virtual device. The "refDevice" Property identifies the physical device on which the service will be deployed, whose coordinates are known in the "location" property. The "deploymentInfo" property contains all the pieces of information necessary to retrieve the data for a docker image to be deployed. The properties identify both the provider and the data to download the image. The "provider" property, in addition to providing a readable user name and the address to download the image from, guarantees reliability as it has been extracted from a trusted database. The "dataFunction" property contains the data for the Edge device Service Starter module. The necessary pieces of data are those for identifying the image and for configuring the sensor. In the first case, these are the path to the image on the provider, the script programming language and the source

Docker file. In the second case, it is metadata such as the name of the image on the device, the acquisition frequency, the timeout, the transmission frequency and the operation to be performed. The operation can be the deployment, update or deletion of the service with all related data.

8.2.3 Design

Figure 8.8 shows the blueprint of the whole system architecture aimed at the automatic deployment and execution of MS. The architecture is divided into layers to allow a better description of the various features. The first layer is the User Layer which manages communications with the end user. It is a system that allows the user to enter data and receive feedback. The data entered is used to configure the on-demand service on an Edge device, while the

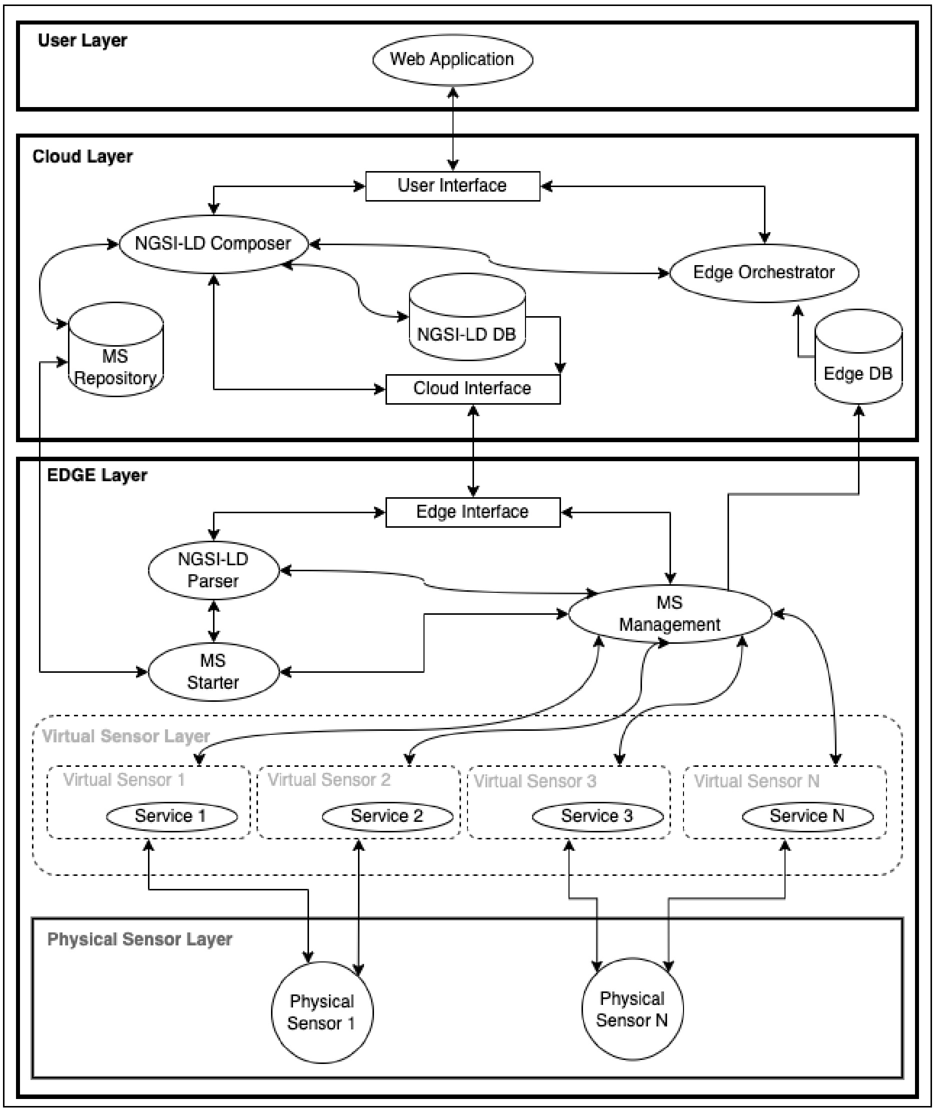


Figure 8.8: System architecture.

information received refers to everything related to the requested service, from the data collected to the status of the service. The Cloud Layer refers to all the software components running in the Cloud, which process the user request and enable communication with the Edge device. It also deals with the processing of information received from the user and their organization according to the NGSI-LD protocol. In addition, the local Cloud ensures the reliability of the microservice image. The repository from which to download the image will be indicated in the NGSI-LD file, and therefore the image will refer to those of the local Cloud. It is assumed that all images in the Cloud have gone through a validation process before being put into production in the smart environment. In detail, the Cloud Layer includes:

- *User Interface*: it is a software module that takes care of conveying information to and from the User Layer;
- *NGSI-LD Composer*: it is the module that, taking user data as input, generates a file in NGSI-LD format to be sent to the Edge device through the Edge Interface. This module, starting from the user information, retrieves the information relating to the image to be used for deploying the microservice and the device on which the microservice must be deployed. Edge device and Image data are pulled from Edge Orchestrator and MS Repository, respectively. The generated file is stored in the NGSI-LD DB;
- *Edge DB*: it is a database containing all the pieces of information on Edge devices;
- *MS Repository*: it is a repository containing all the images to be used for the MS on the Edge Device;
- *NGSI-LD DB*: it is a database that contains the NGSI-LD documents generated in order to always maintain a link between services and users.
- *Edge Orchestrator*: it is a software module that controls the Edge devices. This component reads information to Edge DB which collects information about Edge devices and their status. Edge Orchestrator can also send information to the end user;
- *Cloud Interface:* it is the module that takes care of conveying information to and from the Edge Layer;

Edge Layer describes the architecture of each individual Edge device within the described system. Each Edge device must be able to receive and interpret the NGSI-LD file composed in the Cloud Layer, deploy the microservice and communicate data and/or information to the user through the Cloud Layer. The Edge Layer consists of:

- *Edge Interface*: it is a software module that manages the communications between the Cloud Layer and the Edge Layer. Communications can be bi-directional;
- *NGSI-LD Parser*: is the module that checks and extracts information from the received NGSI-LD document. If a new service is to be deployed, the information is passed to the MS Starter. If the NGSI-LD document contains modification/update information of an existing service, the information is passed to the MS Management;
- *MS Starter*: it is a software module that takes care of running the service requested by the user by deploying a microservice. The MS will be configured according to the data collected from NGSI-LD document. Once the service has been executed, it notifies the service management;
- *MS Management*: it is a software module that monitors the MS deployed on the Edge. The MS Management acts as a broker between the various services and the Cloud to update it on any changes in the status of the services;
- *Service 1,..., Service N*: these are the services requested by users, and are independent MS each of which performs a specific task based on the sensor from which it requests data. Multiple services can refer to the same physical sensor;
- *Physical Sensor Layer*: contains the on-board sensors (Physical Sensor 1, ..., Physical Sensor N) of the Edge device. Each device can host digital or analog sensors that send data to the Edge via the General Purpose Input/Output (GPIO);
- *Virtual Sensor Layer*: This is an abstract component that contains the virtual sensors (Virtual Sensor 1, ..., Virtual Sensor N). The virtual sensors (and therefore the Virtual Sensor Layer) are represented with dotted lines precisely because they represent the concept of the physical sensors for the user but in fact, they are independent MS that process/read data from a shared physical sensor. In fact, they coincide with the Service 1, ..., Service N components which are the real software components which then supply information and data to the users.

The described architecture reports the components for realizing the proposed system. The Edge is to be replicated on each device and each of them can then communicate with the Cloud which instead can be unique. The proposed architecture clearly shows that given a number of physical sensors N installed on an Edge device it is possible to deploy a number of services S, such that $N \leq S$. This assumption demonstrates that it is possible to reduce the number of physical sensors to be used while still guaranteeing a high quality of service. This is possible thanks to the fact that with the deployment of MS, each user will think that he uses his own physical sensor. The number of N sensors installed on the Edge device will depend on the physical limits of the GPIO, while the number S of services (and therefore of virtual sensors) will depend on the hardware characteristics of the Edge device.

8.2.4 Workflow for MS deployment

A key objective of our work is the implementation of an on-demand virtual sensor deployment flow. Figure 8.9 shows the sequential representation of the flow that is essential to the operation of the system described in Section Design. Through the User Interface (Figure 8.9 - Step 1), the Cloud receives the user's request. This request, if valid, is passed to the NGSI-LD Composer. NGSI-LD Composer creates the NGSI-LD document (Figure 8.9 - Step 2). The identification of the physical device takes place through an exchange of data with the Edge Orchestrator (Figure 8.9 - Step 2a). If the requested Edge device exists and is available then NGSI.LD Composer completes the document and sends it to the Cloud Interface (Figure 8.9 - Step 3). The Cloud Interface sends the NGSI-LD document generated in the Cloud via HTTP connection to the Edge device identified to satisfy the user request (Figìure 8.9 - Step 4).

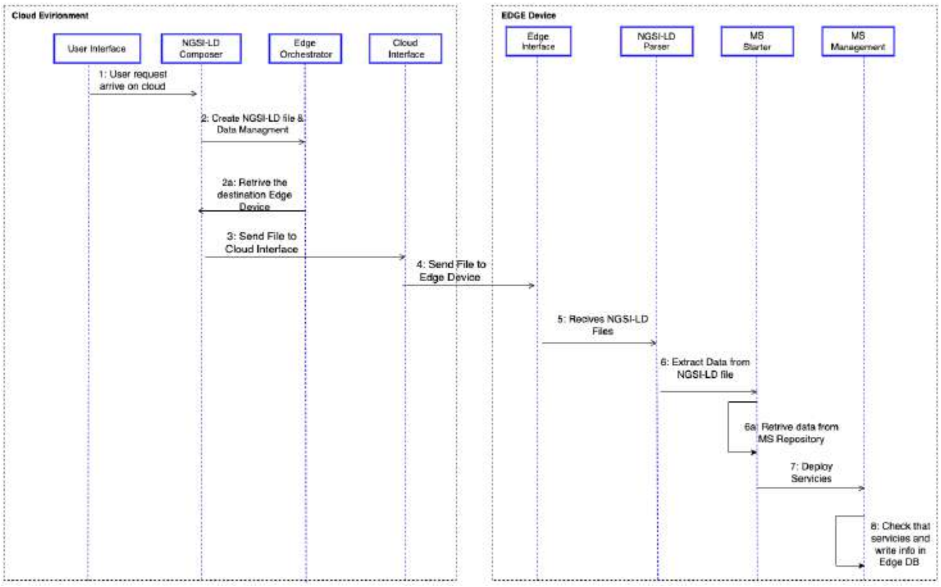

Cloud Evirionment
User Interface
NGSI-LD Composer
Edge Orchestrator
Cloud Interface
1: User request arrive on cloud
2: Create NGSI-LD file & Data Managment
2a: Retrive the destination Edge Device
3: Send File to Cloud Interface
4: Send File to Edge Device
EDGE Device
Edge Interface
NGSI-LD Parser
MS Starter
MS Management
5: Recives NGSI-LD Files
6: Extract Data from NGSI-LD file
6a: Retrive data from MS Repository
7: Deploy Servicies
8: Check that servicies and write info in Edge DB

Figure 8.9: Sequence Diagram.

The document is received by the Edge interface and, after a quick validation check, it is sent to the NGSI-LD Parser (Figure 8.9 - Step 5). NGSI-LD Parser extracts the data needed to deploy the microservice that satisfies the user's request (Figure 8.9 - Step 6). The extracted data is sent to the MS Starter. This component downloads the deployment information from the MS Repository and deploys the microservice (Figure 8.9 - Step 6a). At this point, the control of the service is passed through a handle to the MS Management (Figure 8.9 - Step 7). The MS Management will check the functioning of the microservice and possibly update the status of the Edge Device on the Edge DB (Figure 8.9 - Step 8). Now a microservice on the Edge device performs a function for the user by fetching data from a physical sensor. The user will receive the data and how it does so is outside the scope of this paper. However, it is essential to underline that the user will be convinced that a sensor dedicated to him will provide him with data. In reality, however, the sensor mounted on the Edge device can, using the architecture described, be used independently by N users. These procedures make it possible to significantly reduce the number of sensors used with significant benefits in economic and environmental terms. Furthermore, the system will allow the use of data and services to a greater number of users than in the past. On the one hand, the quality of the service will therefore improve, whereas, on the other hand, Edge devices will be used to their full potential.

8.2.5 Experiments

The experiments carried out had the aim of validating the flow and testing the reliability of the NGSI-LD standard in the described approach. In particular, the ability of a Cloud system to compose an NGSI-LD document, and the ability of an Edge device to extract the information necessary for automatic deployment were tested. Specifically, the Python ngsild client library [243] was used for file composition and parsing. As for the Cloud, an Aruba Cloud Virtual Machine (VM) instance was used with Ubuntu Server 18.04 LTS 64 bit operating system, 8 GB of RAM and Intel(R) Xeon(R) E5-2650L v4 @ 1.70GH CPU (4 Virtual CPUs). It was figured out that this instance could simulate a Cloud. The same test was also carried out on a MAC mini with Chip M1, 8 cores, 16GB of RAM and MAC OS Monterey 12.06 operating system. This represents a private Cloud instance. The Edge device used was a Raspberry Pi4 Model B, 4GB of RAM, Quad-Core ARM-Cortex-A72 processor and Raspian operating system. The tests were conducted with a repetition of 50 times.

Cloud Test

The result of the test on the public Cloud (i.e., an Aruba VM instance) is shown in Figure 8.10. The average execution time needed to create the file that allows automatic deployment is about 0.57 milliseconds. The execution time during the experiments is linear except for some values which can be considered outliers and which do not influence the results. The result of the test on the private Cloud (Mac Mini) is shown in Figure 8.11. The average execution time needed to create the file that allows automatic deployment is about 0.18 milliseconds. Even in this case, the execution time during the experiments is linear except for some values which can be considered outliers and which do not influence the results.

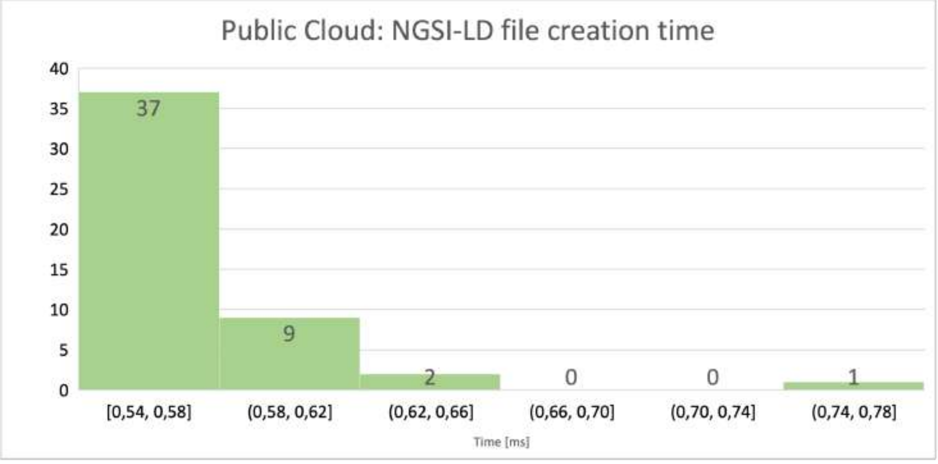


Figure 8.10: Public Cloud experiment.

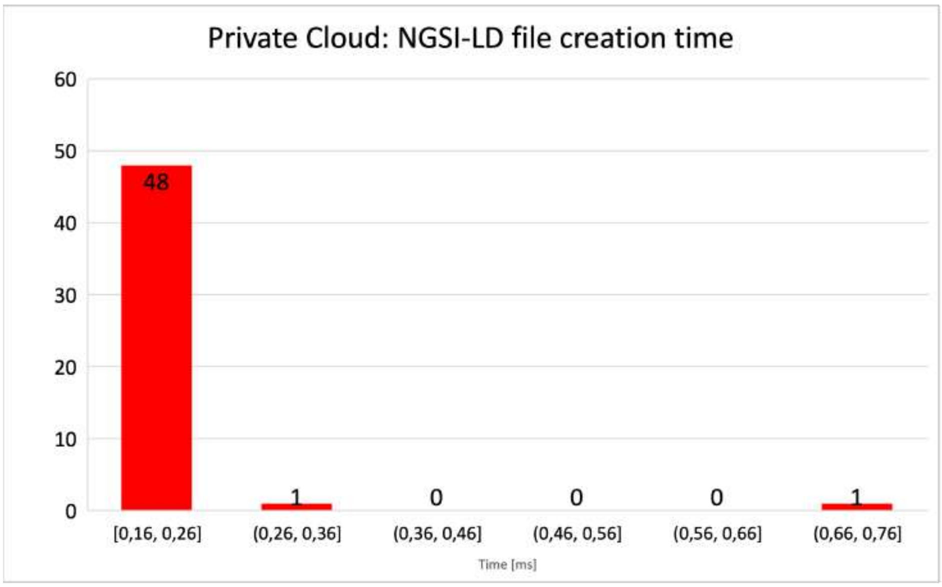


Figure 8.11: Private Cloud experiment.

Figure 8.12 shows a comparison between the average execution times in the two systems. The results clearly show better performance with the device used to simulate a private Cloud. This test demonstrates how important the choice of hardware is at scale to avoid bottlenecks due to document processing. By comparing the hardware used, the reasons for

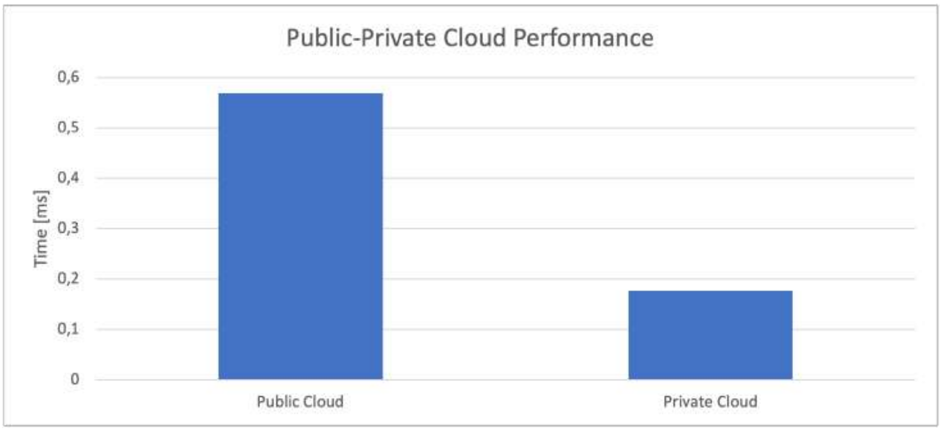


Figure 8.12: Public-Private Cloud comparison.

these differences are clear.

Edge Test

Figure 8.13 depicts the results of tests performed on the Edge device. In this case, only the ability of the device to extract the information necessary for the deployment of the specific service was evaluated. The measure represents a fundamental part of the NGSI-LD Parser component shown in the architecture in Figure 8.8. The average time it takes for the device to extract information for MS deployment is 0.21 milliseconds. The NGSI-LD document contains additional information with respect to the deployment data. However, this information is not necessarily used for deployment. This condition, therefore, allows evaluating the parsing of a specific part of the NGSI-LD document.

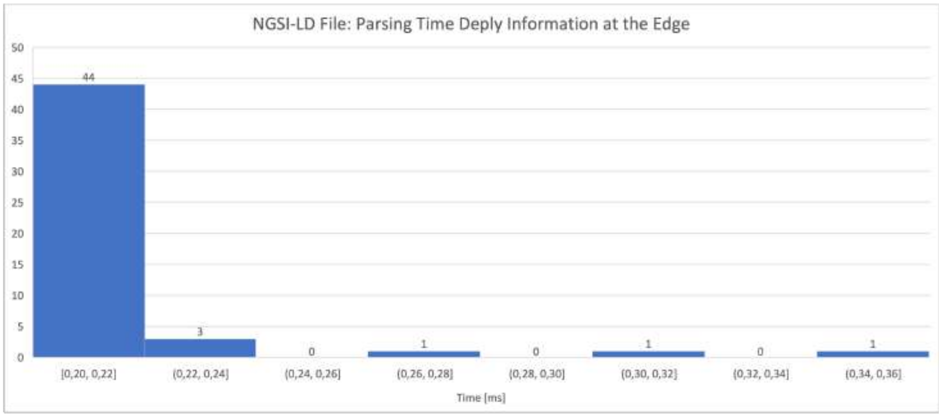


Figure 8.13: Edge device experiment.

8.2.6 Final Remarks

The proposed work introduces an innovative method for the automatic deployment of MS on Edge devices. In particular, the aim is to start from a user request to be able to automatically compose a configuration file which can then be sent to the Edge device which extracts the information. To achieve the objectives, it was functional to use the NGSI-LD protocol which allows us to define customizable files. The described architecture defines the components necessary for the automatic deployment and an example of the structure of the NGSI-LD file necessary for the transmission of the information. In the implementation phase, the implemented components were described to validate the architecture. The flow demonstrates how it is possible to reduce the physical number of sensors while still increasing the number of users who benefit from the data that the sensor collects. The concepts of both Virtual and Physical devices are explained. In the end, in the test phase, the results concerning the validation of the composition of the NGSI-LD file on the Cloud are reported, and the reading of the information necessary for the deployment on the Edge device. The results show that the main components work well reaching the purpose for which have been defined. In the experiments on the Cloud, the importance of the hardware to be used emerges for the purpose of a faster response in terms of computation time by the components. Future work will concern the completion of our system implementation. It will also be necessary to carry out an assessment of the MS scalability, specifically investigating how Edge devices scale up when they have to manage a high flow of requests for on-demand services. Another factor to be tested in the future will concern the scalability of the Cloud in managing a large number of devices on which to deploy on-demand services.

8.3 On-Demand Management for IoT Devices

One of the major drivers of current digital transformation is undoubtedly the Internet of Things (IoT). This technology is revolutionizing our daily life by connecting several smart objects around us providing innovative user-centric services. IoT is then changing the way we interact with the surrounding environment, allowing us to access a huge amount of information and, most of all, improving several aspects of our life, such as logistics, hospitality, healthcare systems, transportations, energy-saving industries and new solutions that need real-time reactions. Many IoT applications follow the logic where four key functionalities are executed cyclically for: (i) collecting data from the environment (sensing), (ii) transfer data to a processing unit (communication), (iii) elaborating data (computation), and (iv) pushing a machine or device to operate (actuation). However, these functionalities are not all implemented into the IoT layer. Sensing and actuation are performed by IoT devices, whereas communication and computation are in charge of more powerful systems often labeled as Edge devices. Their main difference comes from their hardware architectures: (1) IoT devices are often based on MicroController Units (MCUs) and typically used in compact and low-energy equipment (i.e. Arduino and ESP32); (2) Edge devices are instead based on MicroProcessor Units (MPUs) and typically used for low-complexity and low-latency computing purposes (i.e. Raspberry PI 4 and Jetson Nano).

The new generation of IoT infrastructures can do massive usage of both cheap MPU and most powerful MCU devices useful to increase processing and storage resources at the Edge. All these components should be organized for: (i) sharing data (both raw information collected from the surrounding environment and pre-processed ones collected from distributed storage systems); (ii) communicating with each other for maintenance or reconfigurability purposes; (iii) and executing local processing.

Therefore, focusing on IoT based applications, the objective of this paper is discussing a technique to push processing from the traditional Cloud-based distributed systems to interconnected Edge/IoT end-devices, configured into a strongly dynamic mesh network. Another important aspect to consider about the new generation of IoT-based applications is that processing needs change over time. Reactive systems where actuation change the behaviour of the environment need to sense (and then process) data according to the specific conditions. For example, during the monitoring of a smart building, in case of fire, the rate of temperature measurements could be increased to analyze how the fire propagates, or in smart agriculture, the analysis of the soil could change according to the season or plantation. To address these needs, in this paper we investigated heterogeneous Edge/IoT mesh networks, with special attention to the dynamic and on-fly configuration of nodes, with particular regard to MCU ones. Specifically, we adopted the Multi-Hop-Over-The-Air (MH-OTA) protocol to drive the reconfiguration of MCU nodes by a remote message, thus enabling the infrastructure to redeploy services whenever it is necessary. The work presented allows the deploying of "ad-hoc" services in extreme conditions for connectivity. The study conducted allows us to demonstrate that it is possible to reprogram MCU devices even in rural areas or in dangerous situations, where not all devices, even in normal work state, are able to have connectivity. This factor is innovative compared to the state of the art, and opens up new scenarios for the development of Edge-IoT applications.

8.3.1 State of the Art

In the recent past, many attempts to build IoT systems adopting customized approaches had mostly failed due to the intrinsic complexity of services deployed on distributed interconnected devices. In this context, the main building blocks of a general purpose IoT platform called Mainflux are discussed in [244]. Mainflux is built according to a microservice deployment style using a container virtualization approach. Binding is based on HTTP, MQTT, WebSockets and CoAP technologies. Although such a platform is interesting, it does not take advantage of the mesh network in terms of faster broadband, seamless roaming and setup simplicity. In fact, the advantages of hybrid mesh networks have been widely investigated in the literature. A survey of the relevant technologies that may be suitable for mesh networking, either providing native support or being adapted subsequently was discussed in [245]. From the application point of view, it also identifies several application scenarios that can benefit from the deployment of hybrid mesh networks. A piece of framework called "Wireless-Fog Mesh" for in-network computing of microservices in dynamic smart environments is discussed in [246]. Such a solution is based on a fog controller that exploits underutilized resources of mesh devices and network metrics for mapping a microservice to a fog node. The system is based on a DHT overlay of fog nodes also acting as distributed MQTT broker for disseminating data to fog nodes. A wireless IoT hardware platform that uses LoRa as a communication protocol and is able to connect and manage several types of sensors was presented in [247]. Then, they investigate the advantages of combining into a mesh network devices with wide-area coverage (by means of LoRa) and devices characterized by ultralow-power consumption WPAN protocol (by means of the ANT technology). The cooperation of different IoT technologies in industrial environments, through a collaborative mesh network based on Bluetooth low energy (BLE) and long range wide-area network (LoRaWAN) is encouraged in [248]. In the proposed architecture, the IoT system is connected to a fog server responsible for preprocessing raw data that will be stored in a global cloud server, being available for consults at any time. In this paper, however, IoT devices are configured only to sense context data and send it to a back-end through the gateway for storage and processing. An IoT mesh system is investigated in [249] where the physical network combines two subsystems: a Bluetooth Low Energy Mesh network of boards equipped with sensors, and a security monitoring system equipped with passive infrared sensors and cameras. Even if the paper presents an interesting implementation of a heterogeneous IoT infrastructure, the behavior of IoT devices is static and configured a priori. Moving towards fully distributed approaches, [250] proposes a generalized framework for in-network computation, in which the data aggregation and processing steps are implemented through an artificial neural network distributed across the IoT mesh network. This work overcomes the limits of the in-network computation literature that focuses on simple functions. However, the required operations on the input data collectively perform by a predefined number of IoT neurons. The emphasis of this contribution is related to the data flow model for processing instead of the reconfigurability of processing tasks. A lightweight virtualization, Information-Centric Networking (ICN), and service deployment algorithms to facilitate efficient service delivery in Community Mesh Networks, which are decentralized mesh networks built to satisfy the community's demand for Internet access and to provide services of local interest is discussed in [251]. The proposed decision engine selects the appropriate nodes for service instantiation based on constraints observed in network bandwidth, available hardware resources and network topology. To distribute the services over the network, the engine manages Docker containers which can be easily deployed at the edge and make use of the Named Data Networking solutions to distribute in-network container caching without any control entity. A microservice architecture model suitable for building IoT applications is discussed in [252]. Applications are composed of loosely coupled microservices. Such microservices are packed in Docker containers, deployed over a mesh of devices and the plenty of containers related to the same application are orchestrated using Kubernetes. These solutions are challenging but not applicable if there are MCU devices that can not hold containers. The remote and automatic configuration of IoT devices especially in large networks is an open issue. In literature, some solutions deal with the automatic configuration of IoT devices. For example, an IoT protocol for the autoconfiguration of IoT devices in smart environments such as houses is discussed in [253]. However, the protocol addresses only the initial configuration of devices at the service setup stage. About the reconfiguration of IoTs devices and applications over-the-air (OTA), the evaluation of some solutions for the well-known embedded device Telosb with TinyOS is provided in [254]. TinyOS provides the necessary abstractions useful to install different tools such as Mires, Deluge, TinyLIME and so on. On the contrary, in our work, we operate at a lower layer considering devices that cannot execute an operating system. In our approach, the OTA protocol drives the reconfiguration of MCU nodes by a remote message.

8.3.2 Materials and Methods

This work aims at leveraging a MPU-MCU mesh network acting at the Edge with particular regard to the configuration of MCU nodes. Specifically, this Section describes the different elements that were designed and deployed to enable the MH-OTA firmware update of all MCU nodes within the mesh network deployed on the Edge. Moreover, it will highlight the innovative aspects of our scientific work, that are:

- the distributed firmware Edge storage repository;
- the OTA update of several mesh network nodes driven by human needs.

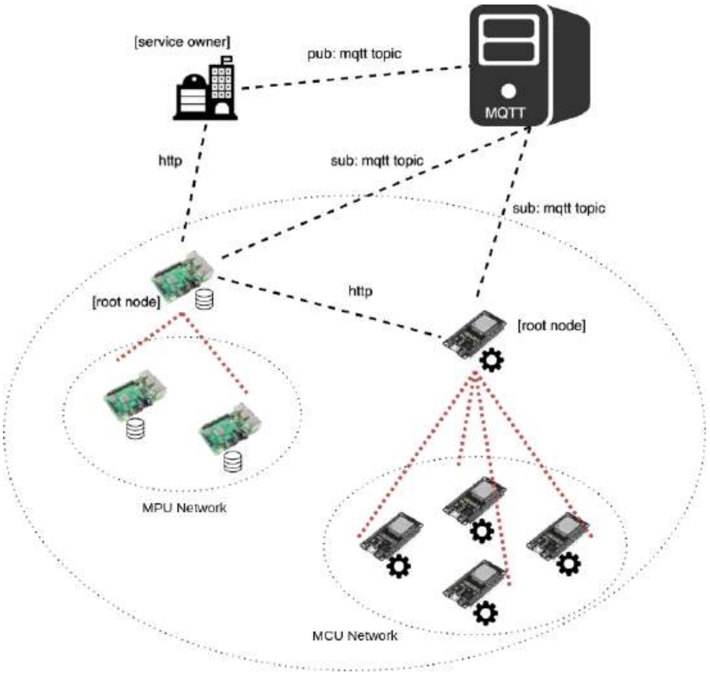


Figure 8.14: Infrastructure overview.

Figure 8.14 shows the infrastructure overview of the proposed solution. Two mesh networks are firstly deployed on the Edge. One of them is dedicated to store firmware. We have decided to use MPU devices, such as Raspberry Pi 4, for accomplishing this goal and serving the firmware over the private network by means of a server web. The Raspberry Pi 4 root node communicates with the root node of the second mesh network, which is dedicated to computation. In this case, the network is composed of MCU devices, such as ESP32 system-on-chips. Both networks elect the root nodes from time to time, allowing communication between them. The full cycle composed of sensing, communication, computation and actuation is fully functional over these two mesh networks.

A message protocol is instead used for enabling communication between the mesh networks and the service owner. A new firmware is then uploaded into the MPU Edge network and served by server web; they are so installed via Multi-Hop-Over-The-Air (MH-OTA) protocol on all MCU nodes, driven by a remote message. Our system, therefore, enables a full re-configuration of the MCU network, deploying new services remotely whenever is needed.

Distributed Firmware Repository

To allow data sharing between cluster nodes, GlusterFs has been chosen as a distributed and scalable Open-source volume management solution, thanks to the data maintenance and saving features in case of failures, using a unique hashtag for each file, stored within the file system itself. To use GlusterFS, it is required that each node, being this a client or a server node, has the service installed. Server nodes maintain the data in the form of volumes and act as a storage pool to expose the directories. Each client can then configure the service which is seen as a normal mounted volume by the operative system. GlusterFS supports various types of volumes as needed. Some types are suitable for scaling storage sizes, others for improving performance, and still others for optimizing both of them. Edge devices can benefit from the concept of High-Availability: a volume can be configured as a replica, to reduce the data loss risk, as copies are provided by each node. For example, with a configuration set on "replica 3" directive, each file is written three times across the nodes. Volumes can be configured as distributed when scalability is crucial and data loss is acceptable. To improve data redundancy, high availability and high reliability there are several options: Replicated, Distributed replicated, Dispersed and Distributed dispersed. For implementation details, please refer to official documentation [255].

For the scope of this research, all pros and cons have been carefully analyzed and it has been chosen the Distributed replicated approach, keeping in mind that the firmware file size to store is about 1 MegaByte, hence required storage space is easy to be obtained, and the critical requirement is to guarantee that the firmware is always available to be downloaded by IoT devices.

MPU-MCU Mesh Network Communication

The idea of updating the mesh network from the long-distance is of great importance when it is deployed on hard-to-reach locations. In this case, service owners need a reliable communication strategy for reaching the desired destination. As a consequence, we have designed to use a message protocol for exchanging information among the mesh and the external networks. MQTT is an OASIS standard messaging protocol designed for IoT integration. It is extremely lightweight and uses the publish/subscribe messaging transport, which is ideal for connecting remote devices with a small code footprint and minimal network bandwidth. It is low-energy and compared to RESTful protocol, MQTT helps to consume less power[3], turning it into a optimal fit for IoT devices. Moreover, a well-designed set of topics leads to good scalability policies.

Design choices have led us to create a topic based on the mesh network ID in the form of: */MESH-UUID/ota/update*. Proceeding in this way we have the opportunity to reach out to the root node of the specific mesh network, scaling over multiple ones whenever other networks are deployed. The payload is made standard thanks to a JavaScript Object Notation (JSON) encoding, allowing passing multiple arguments. Remembering that both the storage-based mesh network and the computation-based mesh network are located on the same private network, the endpoint sent over the message protocol is only accessible by them-self.

Adoption of Smart MCU End-Devices

As already mentioned before, this work aims at giving more responsibility to IoT end-devices for enabling the full cycle (sensing, networking, computation and actuation). The MCU choice was then fundamental for having a low energy device with good computing capacity. Moreover, taking in mind the wish of remotely injecting new services, software development and deployment flexibility was also strongly required. Correspondingly, we have chosen to use the Espressif Systems' ESP32 as an end-device, which is a low-cost, low-power system-on-chip microcontroller with integrated Wi-Fi and dual-mode Bluetooth. ESP32-WROOM-32 contains two low-power Xtensa 32-bit LX6 microprocessors individually controlled, and the CPU clock frequency is adjustable from 80 MHz to 240 MHz. The chip has also a low-power co-processor that can be used instead of the CPU to save power while performing tasks that do not require much computing power. All software components run on freeRTOS, i.e., a free real-time operating system with a lightweight open-source TCP/IP stack, such as LwIP.

Each MCU end device is part of a mesh network built atop the WiFi protocol. The mesh nodes simultaneously act both as the access point for enabling multiple downstream connections and a station for maintaining a single upstream connection, resulting in a tree network topology with a parent-child hierarchy consisting of multiple layers. Each node can transmit packets to other nodes through one or multiple hops but simultaneously serves as relays for other nodes. Therefore, the network configuration process becomes independent of routers.

The implementation we have built on top of the Espressif's Mesh Development Framework (ESP-MDF) follows the automatic root node election via the physical layer protocol.

[3] www.bevywise.com/blog/mqtt-vs-rest-iot-implementation

Indeed, when devices start up in the network, they look for others in the WiFi range. The network is then built layer by layer, starting from the root node, which will be elected according to the received WiFi power signal. Such a node serves as the only interface between the mesh network and the external IP network, especially the repository server web. The self-organizing property of the network lets itself elect a new root node when this is detected as broken.

Firmware Injection into MCU Devices using the MH-OTA

The firmware injection makes the whole mesh network re-programmable and suitable for general-purpose applications. Such a mechanism is made possible through MH-OTA updates, which allows an end-device to update itself and other nodes distant one or more hops, according to data received over WiFi in run time. The MCU end devices are configured with a partition table including two OTA partitions: one for applications and one for data. As a result of the MH-OTA update, the OTA data partition is updated to specify which OTA application partition should be booted next. As a fault tolerance system, the implementation has a rollback policy for always keeping the device working. Therefore, if the injected application has critical errors, the rollback policy allows starting the previous firmware version. The MH-OTA update starts from the root node, triggered by a MQTT message sent by the service owner. The Mesh is indeed passive, waiting for an update pushed by the external network. However, it continuously listens for starting the process. The root node then follows the steps below for distributing the firmware over the mesh network:

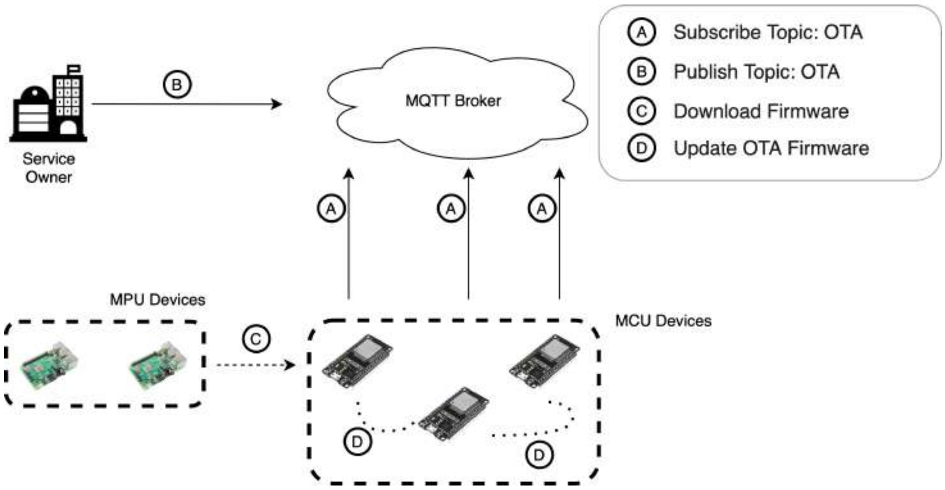


Figure 8.15: Device OTA Firmware update.

1. requires the firmware contacting the server web hosted on the edge

2. writes the firmware on the unused OTA partition
3. splits the firmware in fragments and sends them to all nodes
4. when all fragments are received, asks all nodes to complete the update just restarting

8.3.3 Results and Discussion

The goal of the experiment is to understand how the MH-OTA update performs in terms of time while the involved node is also committed to computational tasks. Therefore, we implemented five firmware versions, in which the nodes perform tasks increasing the operation rate (0.1 ms, 1 ms, 10 ms, 100 ms, 1000 ms). Another firmware deactivates any task, leaving the node idle and representing the benchmark of the experiment. The MH-OTA update message is sent following a task execution period of 10 seconds. All the experiments are repeated 20 times. Through these experiments, we aim to study the Mesh Network behavior during the MH-OTA update while any node executes a digital signal processing (DSP), floating-point arithmetic operations such as sum, subtraction and multiplication. The FFT is one of the most commonly used operations in digital signal processing to provide a frequency spectrum analysis. The Fast Fourier Transform (FFT) computes the Discrete Fourier Transform of input much faster than computing it directly. In other words, the FFT reduces the number of computations needed for a problem of size N from $O(N^2)$ to $O(NlogN)$. [4]

The performance metrics of relevance for the experiments are the following:

average of download time is the time required for downloading the firmware within the root node. It starts when the trigger event happens and it ends when all bytes are downloaded and stored into the OTA application partition. The metrics gathered while the root node performs both the firmware download and a computation task are compared with the metric gathered when the root node is only committed in the downloading process (idle).

average of routing time is the time required to propagate the firmware from the root node to all the mesh network's nodes. It starts when the firmware is downloaded and stored into the OTA application partition and it ends when all firmware's frames are propagated through the mesh network. The metrics gathered while the nodes perform both the firmware propagation and a computation task are compared with the metric gathered when the node is only committed in the routing process (idle).

[4] https://towardsdatascience.com/fast-fourier-transform-937926e591cb

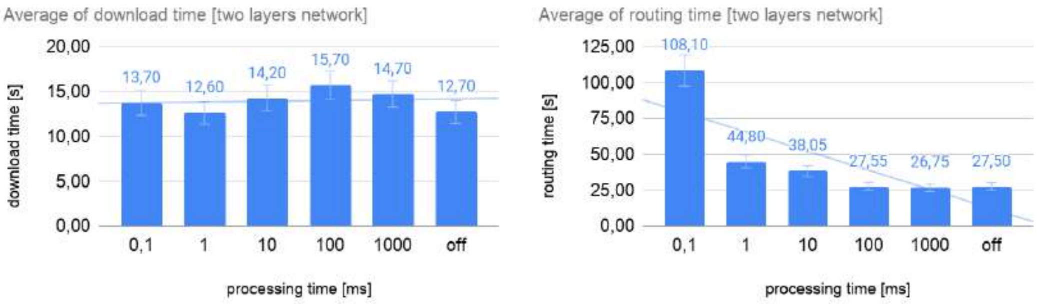
In order to carry out the above-mentioned experiments, the solution described in this paper has been tested over two networks, where we have changed the number of nodes and layers. Both experiments have the same storage mesh network, composed of two devices based on Raspberry Pi 4 Model B with a Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz, 4GB LPDDR4-3200 SDRAM, 2.4 GHz, and 5.0 GHz IEEE 802.11ac wireless. They are configured for using GlusterFS and share a repository of the firmware. The firmware size is 1052576 bytes, and it is served by a simple server web built on Python 3.7. The MQTT broker is installed on a server with an Intel(R) Xeon(R) E-2124G CPU @ 3.40GHz and 32GB SDRAM. The computation mesh network is based on ESP32-WROOM-32 with two low-power Xtensa 32-bit LX6 microprocessors and a lightweight open-source TCP/IP stack. The implementation is fully based on the Espressif Mesh Development Framework, a networking protocol built on top of the Wi-Fi protocol. It runs on a free operating system (freeRTOS) and executes the processes in tasks. That means, only one of them within the application can be executing at any point in time and the real-time RTOS scheduler is responsible for deciding which task this should be.

Two Layers Network

The first tested network is composed of two nodes organized in two layers. One node is elected as root in run-time, whereas the second one is attached as a child. The routing table then follows a direct connection between the two nodes.

Figure 8.16a shows the average download time. On the x-axis, it reports the processing time used over the computation tasks, whereas on the y-axis it reports the download time in seconds. Considering also the confidence interval, the comparison of the download time among any processing time experiments, and mostly with the "off" experiment (when the root node is idle) shows a constant behavior. This means the download time is not affected negatively by the concurrent computation of specific tasks.

Figure 8.16b shows the average routing time. On the x-axis, it also reports the processing time used over the computation tasks, whereas on the y-axis it reports the routing time in seconds. The firmware is then split into chunks and sent over the network. The routing time shown in the Figure 8.16b has an exponential behavior. Indeed, by decreasing the task operation frequency, the routing happens faster. It is interesting to highlight the lower bound that starts with operations at 100 ms and continues with nodes in idle. This means the routing time is not affected by specific tasks when they have a period of operation greater than 100 ms. On the other hand, a real-time task (0,1 ms) affects the timing needed to propagate the



(a)

(b)

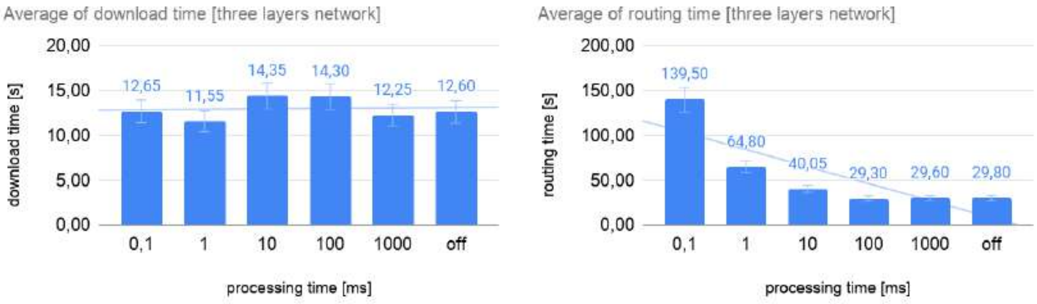
Figure 8.16: a) Average of download time and b) Average of routing time experiments of two layers network.

firmware all over the network.

Three Layers Network

The second tested network is composed of three nodes organized in three layers. One node is elected as root in run-time, whereas the second one is attached as a child. The third node is a leaf of this particular tree, which takes the shape of a chain. The routing table then has a node two hops distant from the root.

Figure 8.17a shows the average download time. On the x-axis, it reports the processing time used over the computation tasks, whereas on the y-axis it reports the download time in seconds. Figure 8.17b shows instead the average routing time. On the x-axis, it also reports the processing time used over the computation tasks, whereas on the y-axis it reports the routing time in seconds. The times shown both in Figures 8.17a and 8.17b confirm the trends highlighted on the previous experiment. The download time is again constant, as the linear approximation line shows in Figure 8.17a. Indeed, just as we expected, the firmware download is independent of the network size and configuration.



(a)

(b)

Figure 8.17: a) Average of download time and b) Average of routing time experiments of three layers network.

On the other hand, although the exponential behavior is still present, the amount of time has increased at any experiment, and mostly in experiments with a shorter period of operation. This means that by increasing the network size, and mostly the number of layers, we might expect the scalability deterioration. In this case, is important acting with suitable routing algorithms and topology. On the other hand, the lower routing time gathered when the nodes have a period of operation greater than 100 ms is a good starting point for building re-programmable nodes without interrupting the computation and staying within an acceptable time.

8.3.4 Final Remarks

In this section, we have discussed how the recent advances in MCU computing power have changed the way we think about the end devices. They are usually labeled as non-smart IoT devices, confining their capacity within the sensing and actuation features. However, such devices are much more. The computing capability has increased over time, arriving at enabling inference on MCU. As a consequence, we have aimed to give more responsibility to them. The approach proposed on these pages organizes the end-devices in mesh networks, a solution of collaborative routing that exploits their potential. In particular, we have seen what are the needs of a network deployed on inconvenient or hazardous locations, where the human cannot easily access the devices anymore and therefore change their behavior using the classic way. On the other hand, the experiments carried on a mesh network based on ESP32 system-on-chip supported the thesis that such devices are able to perform specific operations while their behavior might be changed over time. The injection of the new firmware is indeed efficient in terms of routing time, considering operations executed with a frequency less than 10Hz.

About future works, we would start selecting the Artificial Intelligence techniques that may lead us to validate the self-configuration property. Likewise, we aim to unify the mesh networks used in this paper, including devices of a different kind, such as MPUs and MCUs, in just one. Moreover, we think that a study of topology (star, ring, Von Neumann, etc.) could identify the best social structure the mesh network should take on to minimize the routing time.

CHAPTER 9

Services Optimization in Distributed Environments

Managing services on devices with limited processing capacity requires that they be optimized. This problem is found in Smart City applications ranging from smart mobility to e-heath, etc. One of the use cases where the problem is most relevant is video surveillance in Smart Cities. There are public and private video surveillance systems, and very often different or unique systems the devices frame the same area. However, when an objective requires it to be identified or must be monitored in real-time, such solutions typically require human intervention to configure the devices in the best possible way (for example, choosing the optimal cameras, setting the focus, and so on). This chapter describes a new query method based on a Federated Edge approach. This approach solves the problem from the point of view of both camera hardware and the shooting angle associated with it. According to the presented approach, it is possible to figure out which is the best camera to identify a target and possibly monitor it in a specific area. A case study is defined in the context of urban mobility management. Another interesting approach for optimizing complex services is Federated Learning. This technique allows the training of models related to Machine Learning and mainly exploits the Edge Computing paradigm for training data acquired from the surrounding environment. The solution described in this chapter aims to optimize all processes involved within a Federated Learning client through transparent scalability across different devices. The proposed architecture and implementation abstracts the Federated Learning client architecture to create a transparent cluster capable of optimizing the complicated calculation and aggregation data to solve the problem of heterogeneous data distribution in federated learning applications. Video surveillance but also filming for traffic analysis in the smart city causes privacy problems as well as the cost of video data transmission. It is also not easy to build object detection models on large training datasets stored centrally following current approaches in the literature. Federated learning (FL) is a promising approach in this context. From the study of the state of the art, the real value of using these approaches was revealed [256]. The contribution of this chapter proposes improved solutions to the aspects already addressed by the scientific community. [256]

9.1 Federated Edge for Service Optimization

The Smart City model is now a constant in continuous growth all over the world. This sector is the subject of several funding programs in Europe, e.g., Next Generation EU and Horizon Europe make available hundreds of millions of euros for the digitization of services and territories. Particular importance in this digitization process is given to the safety of citizens and the protection of the environment. To this end, the massive deployment of city monitoring cameras is assuming considerable importance. Typically, cameras in Smart Cities are used for video surveillance or traffic control and over the years they have been installed in specific sites for the acquisition of images and videos on a specific road/area. However, modern cameras have network and computation capabilities that make them smart devices able to perform analysis on the Edge in a collaborative way. To better understand the new challenging scenarios for smart city video monitoring, let us consider a group of cameras that film the same place and the same scenes from different perspectives and a reference target that is of interest for the monitoring system (e.g., a vehicle entering a limited-traffic zone, a crowd of people, a too fast mobile object, etc.). The cameras can provide information on the same target with different quality levels that depend on several factors, such as the distance of the target from the camera lens, the camera resolution, the focus angle, the type of target, the velocity of the target, etc., and it is not possible to know in advance which the camera that better suits the monitoring/tracking of the target is. For this reason, we can no more analyze video flows coming from different cameras independently of each other, but we should start thinking about connected systems where a federation of cameras that acquire data on the same target cooperate to provide the highest quality information to the application layer (e.g., video surveillance or traffic control). This section intends to introduce the concept of *Federate Edge*, enabling a dynamic set up of smart cameras on the basis of their ability to recognize a target in a common scene. We start from the characterization of the target itself and the ability of cameras to get information on it. To this aim, we investigate the solid angle construction around the target to normalize data and estimate information quality. Then, we provide design strategies and enabling technologies for the establishment of the Federated Edge. Inside the federation, the devices will collaboratively identify the one that is most suitable for recognizing the target of interest. The system thus designed is able to respond to target-based queries optimizing the results on the basis of data from the best device. In the presentation of the concepts, we will refer to two use cases as concrete examples about the possible benefits of the proposed solution, which are video surveillance and traffic monitoring. They are analyzed with reference to the real deployment of traffic cameras in the city of Messina, which has hundreds of cameras managed by a centralized Video Management System (VMS) and is currently involved in two project on urban mobility, which are the MeSmart project, funded by the Italian PON Metro 2014-2020, and URBANITE funded by the European Horizon 2020.

9.1.1 State of the Art

When different cameras film the same scene, its elements can be observed from different angles, and therefore a camera can recognize the same element or not. In [257] the authors address these problems by basing their study on the recognition through zooming of objects at a great distance. However, this method has limitations in adapting to the scene changes that our approach aims to address. An interesting approach to target recognition in an area is described in [258]. The authors demonstrate that they achieve high accuracy in detecting and recognizing a static target, while this is not the case for moving targets. Our work aims at solving this problem taking into account the devices "close" to the Edge device that locks the target. In [259] the authors introduce an image analysis method that maintains high detection performance by reducing the number of pixels processed by about 70% and the detection time by more than 50%. In our solution we propose how it is possible to carry out analyzes and queries on the Edge reducing the number of pixel acquired. [260] contains important considerations regarding the hardware used in the cameras. In our work we want to propose an Edge-Based model [1] considering the NGSI-LD standard that allows us to choose the camera to be used also considering the on-board hardware. The use of a database to be queried to search for the best camera for the required need also implies the introduction of security concepts [2] and unique geo-referencing of both the camera and the image [261]. Several approaches also based on neural networks for target recognition have been used in the development of computer vision. In [262] some representative target detection algorithms are analyzed considering the problems of algorithms. The study we are proposing aims at finding a solution that can be applied on the Edge by solving the various problems faced in the literature. An interesting study on the recognition of targets in low resolution conditions is reported in [263]. The authors achieved good results. However, it is interesting to compare this method in terms of computing power used with the one proposed in this paper. In [264] it is highlighted how it is possible to use customized generic Edge devices to carry out multiple activities simultaneously as a solution to lighten the work of Cloud infrastructures. The paper merely shows how a target identification algorithm can be run on an Edge computing device. In our work we want to introduce the concept linked to this type of device of being able to be interrogated to understand which targets "see best". Collaborative Cloud and object tracking Edge are presented in [265], in which Machine Learning (ML) algorithms are described in the approach of a partial processing of the video capture on the Edge. Perimeter networks created between devices are used in cooperative cache and video features in [266] where opportunistic algorithms for sharing video portions are taken into consideration. Edge Computing-based adaptive wireless video streaming mentioned in [267] is based on the idea of adopting Dynamic Adaptive Streaming over HTTP (DASH) for perimeter transcoding by cooperating with Edge device and backend. Our goal is to use the Edge approach integrated with NGSI-LD to characterize the device and understand in real time which device is better than another in recognizing and searching for targets even in real-time.

9.1.2 Smart Camera Data Acquisition and Target Identification

The identification of a target in a VMS is a well-known problem in the field of computer vision. The images registered by a camera have several features that could be gathered and analyzed, such as:

- *Scene*: it is the image that represents the whole environment at a specific instant of time;
- *Object(s)*: it is an element in the scene (e.g., a vehicle, a person, a tree, ...) that can be characterized by specific properties (e.g., size, color, movement,...);
- *Target*: it is the "interesting" feature of the scene. The adjective "interesting" depends on the specific use case and, hence, the target is specified by the application/end user and can change during the time. The target can coincide with an object of the scene or can be an information get from a part of the scene (e.g., the movement of an object).

The installation of the cameras depends on the width of the area to monitor and the perspective. Thus, often cameras film the same scene but with different visions. Also, their ability to identify targets is very different due to their specific technological features. For this reason we define *the ability to recognize a specific target of an Edge Device* as the set of its hardware and software properties and its relative position with respect to the target that make it able to provide information on the target. The *quality of information* provided by and Edge device on a target is the probability that the Edge device recognizes the target in one point of the scene is greater than the probability that another Edge device that frames the same scene recognizes the same target. Moreover, the target characterizes the smart camera processing at the Edge. For example, in the management of limited access traffic zones, the target could be the vehicles' license plate and smart cameras have to perform license plate recognition. For this reason, in this paper, we refer to smart cameras also as Edge devices. In the field of smart cities, where there are often hundreds of video cameras deployed in the environment, it may be useful to understand:

1. given a uniquely identified area: which camera has the best capacity to recognize a specific target?
2. given a target, which is the best camera able to recognize it and at which quality level?

The proposed work aims at identifying the methods and technologies useful for setting up an Information and Communication Technology (ICT) system that can answer these questions. In particular, we aim at designing a digital solution which allows querying the system providing as input a target or an area of the city or a specific camera feature, and receives as response data from the camera(s) that best meets the input requirement(s).

Video Surveillance Scenarios

In smart cities, video surveillance is often exploited for urban security and for traffic control. Compared to the possibility of having different cameras that film the same area, there are therefore 2 scenarios.

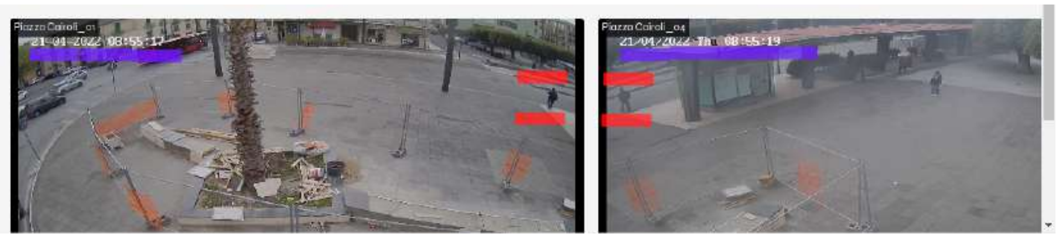


Figure 9.1: Cameras that frame overlapping areas in a square (Piazza Cairoli, Messina-Italy).

The first scenario concerned the urban security. As can be seen in Figure 9.1 in a square, or in an area closed to traffic, it is possible that several cameras film common areas. After the position or even the hardware characteristics it is possible that the analysis capacity of the video is different and therefore a target can be recognized better from one shooting point than another. In Figure 9.1, it is evident how the two cameras partially film the same area with a person crossing the square. If, for example, the person is a target depending from his movements, from the light and also from the presence of any other object in the scene, it may happen that the same algorithm can have different results on one video instead of the other.

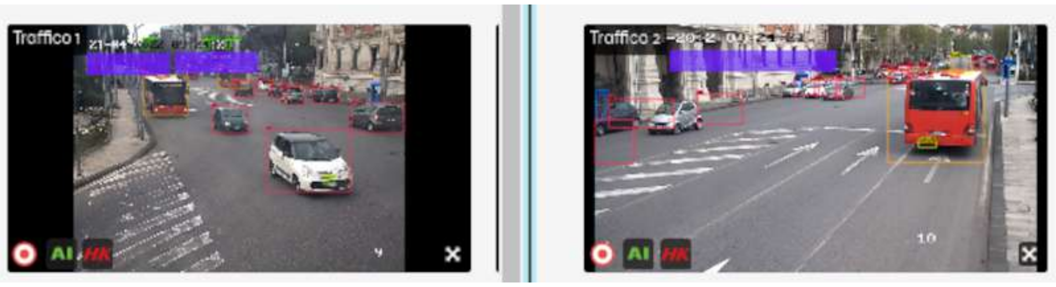


Figure 9.2: Cameras that film overlapping areas in Street Viale Garibaldi, Messina-Italy.

The second scenario of interest concerns traffic monitoring. Also in this case there are situations in which several cameras view the same area. Figure 9.2 shows an example of this scenario. The framed area is located in street Garibaldi, Messina (Italy) and represents a critical artery for city traffic. The 2 cameras frame a large common area from 2 different perspectives, and it is clear that, if the target is the bus, the details captured from the 2 cameras are different. Being a trafficked area it is likely that in some cases it will not be possible to read a vehicle license plate or to clearly understand the details of an accident. If, on the other hand, a system is defined that can modify the type of shot on a point on the basis of a dynamically specification, it is possible to overcome the problems relating to these types of situations.

Another example of the second scenario described is shown in Figure 9.3. In this case even 3 cameras frame a vast common scene located in Vittorio Emanuele II street, Messina (Italy) The framed area is close to that shown in Figure 9.2 and is also a critical area from the point of view of city traffic. It is evident from the scenes in Figure 9.3 that the common area for the 3 cameras is very large. The box identified as "Traffico 7" and "Traffico 6" clearly frame the same truck in the foreground. In "Traffico 7" a vehicle is highlighted in red in the background that you see in the "Traffico 5" box. The same tram stop is evident in the different shots. In traffic conditions, the specific functions of a camera could also be affected here by unforeseen obstacles, particular lighting or traffic conditions. Therefore, the importance of

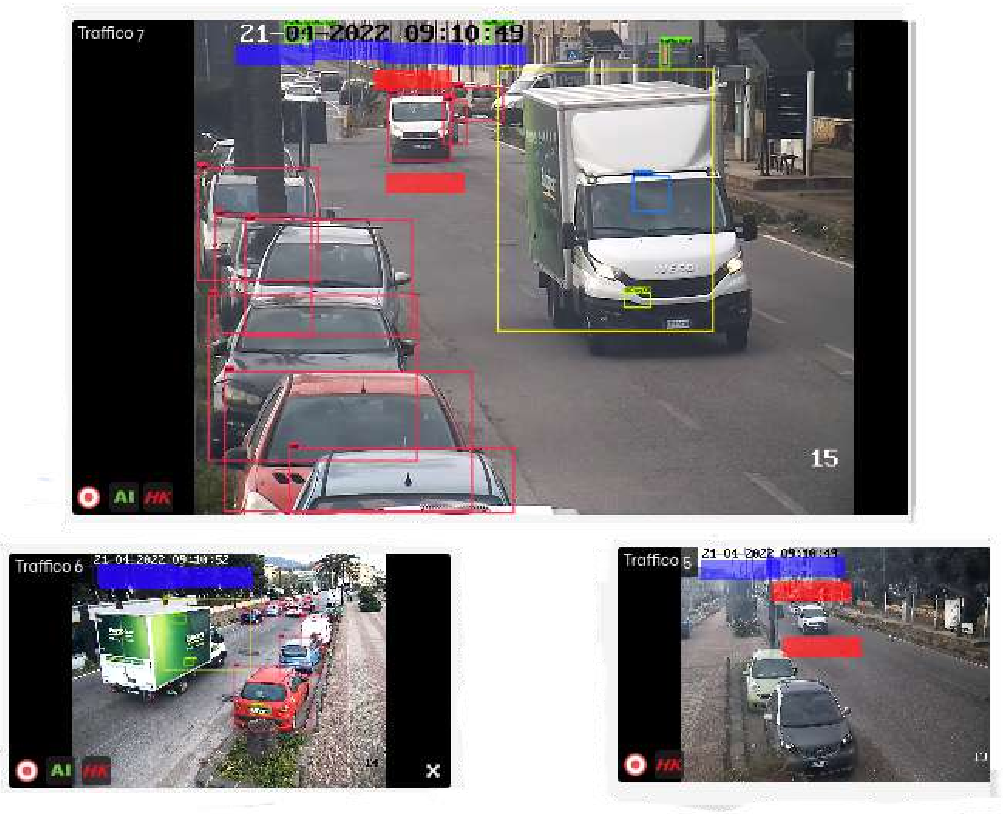


Figure 9.3: Cameras that frame overlapping areas in Street Vittorio Emanuele II, Messina-Italy (Traffico 5, Traffico 6, Traffico 7).

an adaptive system that is able to modify itself according to the requests or specific needs is always greater as the number of devices that frame a common area increases.

9.1.3 Target Tracking for Smart Video Surveillance

Target tracking is an issue related to the effective adoption of image recognition methods. Examples of such interesting methods are described in [268] and [269]. In particular in [269], the parameters that can be used are associated with the solid angle that subtends at the camera lens by the frame of the image.

Solid angle is an extension in three-dimensional space of the plane angle. If we consider Figure 9.4[1] we can define it as:

$$\Omega = A/R^2(SolidAngle)$$

In this formula, A represents the area of the spherical portion of radius R seen under the angle. The ratio, even in the three dimensions, between the portion of circumference, the radius and the subtended angle is maintained. As in the planar angle, the solid angle can be

[1] Source: `http://pngegg.com`

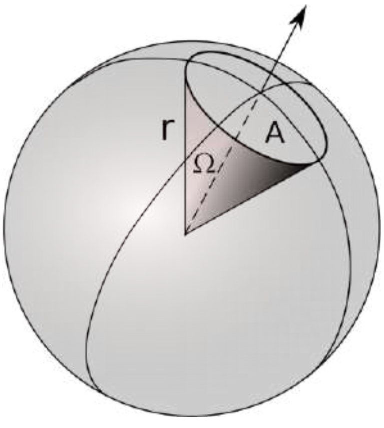


Figure 9.4: Solid Angle representation.

defined how the ratio between the area of the spherical surface and the radius of the sphere considered. For better understand the concept, we can image a light bulb in the centre of a sphere in Figure 9.4. For the whole sphere the solid angle through which the light rays pass is valid. If we consider the only part of the spherical surface crossed by the light rays, using the differentials we obtain:

$$d\Omega = dS/(R^2) = (R^2\theta d\theta d\phi)/R^2 = sin\theta d\theta d\phi$$

where θ is the *latitude* (angle from the north pole) and ϕ is the *longitude*. This value represent the portion of surface that a given camera can be able to frame given a specific angle.

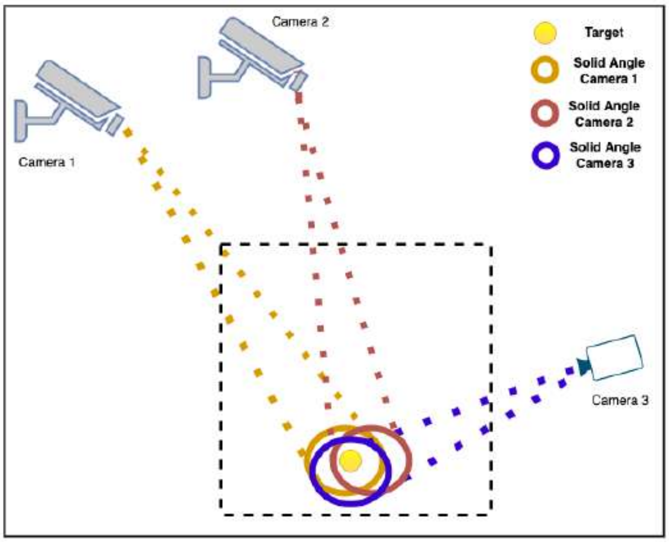


Figure 9.5: Example of Federated Edge with Solid Angle Application.

Figure 9.5 shows a descriptive picture of the reference scenarios. The diagram shows 3 cameras that film the same area identified by the dotted square with their respective solid angles on the reference target. The first parameters that influence the "goodness" of the shot and therefore the ability of the single device to recognize a target are the hardware characteristics and the distance of the Edge device from the target. Each device is able to calculate the distance from the target starting from its position, subsequently the federated devices can elect the "closest" device to the target to be recognized. However, this approach is not sufficient to solve the problems identified. Another parameter that becomes fundamental for our purpose is the solid angle. In [269] the mathematical laws to calculate it are reported, and it is explained how in combination with the other parameters it can be used to better recognize an image. In Figure 9.5 the solid angle is shown with different colors for each camera. The resulting "cone" identifies a particular area of the image on which the device can work. The fact that a target is in the solid angle of the 3 devices allows us to make an exhaustive comparison between their inability to identify it. This comparison can be made at the Edge because analyzing only the pixels in the solid angle greatly reduces the complexity of the image to be analyzed. Therefore, based on the data acquired from the backend system, the devices will know which of them can best identify the target audience. The fact that the devices are federated, and when they talk to each other, still allows for a noticeable improvement. A moving target can in fact change its exposure to the camera. This implies that for various reasons the device defined as "master" at a certain instant of time t understands that one of the devices of the federation can identify the target in a better way and therefore can pass it the title of master.

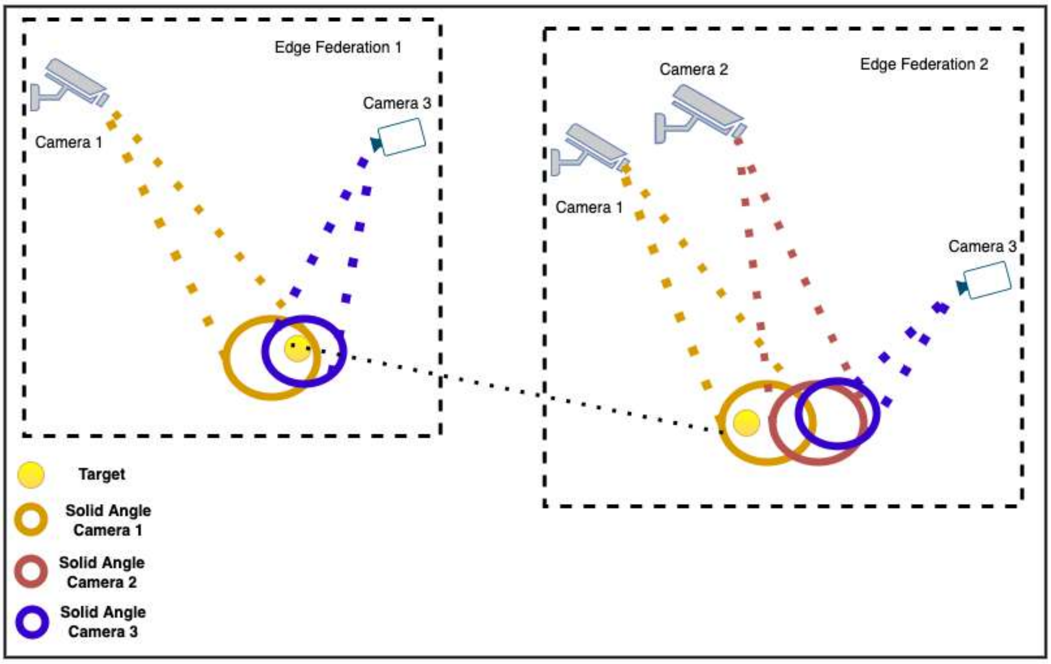


Figure 9.6: Example of Target that move between 2 Federated Edge Environments.

This approach also introduces the possibility of tracking a target within the Smart City. In fact, once the target has been identified and the system knows its position univocally, it is possible that the federation master assigns it a unique label. The consequence of this action is that if the target moves in another group of Federated Edge (but also in the area observed by a single device) it is clearly identified. The situation described is schematized in Figure 9.6. The diagram shows a target that is initially identified in the solid angle of an Edge device that is part of a federation identified as "Edge Federation 1". Subsequently the target moves to the solid angle of another camera located in another federation of Edge identified as "Edge Federation 2". The ICT system described the Design section is able to keep track of the identified target and therefore allow the querying for its identification both in real time and in deferred time. This observation automates the video analysis processes carried out by humans to follow the movements of a target. Furthermore, the data collection means that the querying, for example of a license plate, allows to immediately obtain the positions and/or the routes taken without the need for long video analyzes to be carried out with human intervention. It is evident that the system described is linked to variable parameters. Furthermore, constant communication is required between the different federated devices and with the backend system. To manage this need it is necessary to clearly define the technology and the data model to be used. In this sense, a study has already been carried out in [1]. The NGSI-LD model that is described lends itself to configurations that can be defined for specific use cases and above all allows dynamic modification of the software on Edge devices as well. The definition of the models can be made preliminary and therefore the various needs encountered can be codified by making use of the NGSI-LD to find practical application.

9.1.4 Design Strategies for Federated Edge

Figure 10.1 shows a reference architecture for the implementation of the Federated Edge concept we presented in this paper. The fundamental elements of the system are the ***Edge devices***. They are computational nodes able to process video frames according to the specific requirements of the application in execution for the end user. They can correspond to smart cameras, if these are available and equipped with the necessary software for data analysis, or they can be implemented physically coupling a surveillance camera with an embedded device for data processing at the Edge.

A group of Edge Devices involved in the tracking of the same target forms a ***Federated***

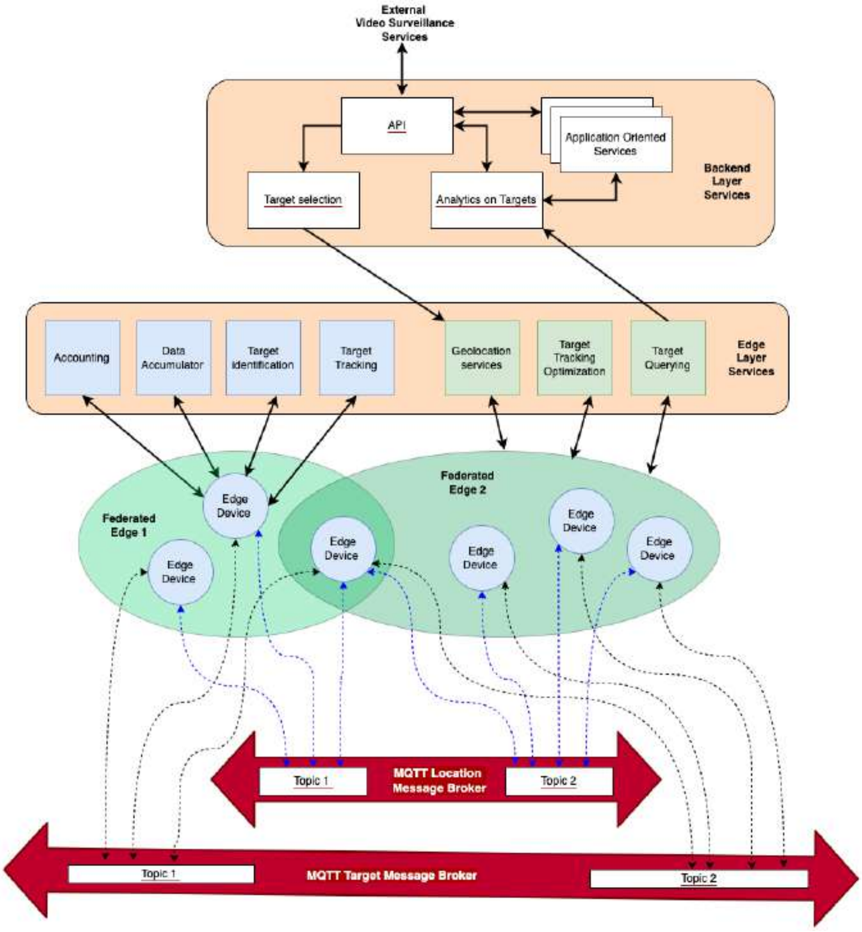


Figure 9.7: General System Architecture.

Edge. Each Federated Edge is configured dynamically on the basis of the target inputs received from the application and Edge Nodes in the Federated Edge cooperate to provide the more significant information to the application. This means that not necessarily all the Edge Devices are part of a Federated Edge and, vice versa, one or more Edge Devices can take part to different Federated Edges at the same time if they are involved in the tracking of different targets in the same scene. In the Federated Edge, it is necessary that each device can know (if it exists) "who is" the close device which shares the same scene and target. So, the device must be able to communicate with its neighbors [5]. To allow the Edge Devices to collaborate in the federation, it is necessary enabling a scalable and secure communication among them and exchange of data. To this aim, the Federated Edge exploit the Message Queuing Telemetry Transport (MQTT) standard, where topics are set up to allows devices to share information. In order to be part of a Federation, a device must:

- Physically share, through localization, the scene in which the target must be identified and then subscribe to a topic of the MQTT Location Message Broker (blue dotted arrow in Figure 10.1);

- Have the ability to recognize the target sought and then subscribe to a topic related to that target in the MQTT Target Message Broker (black dotted arrow in Figure 10.1)

The reference architecture is composed of 2 main layers of services: the ***Backend Layer Services*** and ***Edge Layer Services***.

The ***Backend Layer Services*** include the high level services to support external components (e.g., applications for Smart Cities or end-user data access requests) and they rely on the information provided by the Federated Edge. In particular, they include:

- *API*: it defines the API specifications for communicating with the external components, such as *External Video Surveillance Services* or proprietary video surveillance systems;
- *Target Selection*: it elaborates the target that is requested by the system together with the area of interest where the target has to be analyzed. These information are the input for the activities of the Federated Edge.
- *Analytics on Targets*: it collects data from the Federated Edge and implements specific analytics to create added value for the external components asking for target information.
- *Application Oriented Services*: they are additional services for enriching the value of data coming from the Federated Edge, such as classification of targets within the system as well as cross-relation of data on different targets.

The ***Edge Layer Services*** are software components executed at the Edge. They can be distinguished in services executed independently by each Edge Device and in services executed by the Edge Federation in a distributed and collaborative fashion. Services executed by each Edge Device in an independent way with respect to the other Edge nodes are:

- *Accounting*: it represents the component that manages the authentication of Edge devices and users who use the system;
- *Data Accumulator*: it is a component that takes care of the storage of data and/or information that the Edge devices eventually transmit. This component also has the function of interacting with any software modules that need to access the data history;
- *Target Identification*: it represents the target identification system on the Edge;
- *Target Tracking*: it defines the tracking methods of an Edge device within the federation.

Services executed by the Edge Federation in a distributed and collaborative fashion are:

- *Geo-location Services*: it uses the services described in [261] to uniquely identify a point on the earth if this has not already been identified within the system;
- *Target Tracking Optimization*: it optimizes target tracking between different federations. This function is delegated to the Federation Edge device which is in charge of the target;
- *Target Querying*: it searches for devices that can identify the target or that are tracking it at the time of the request;

Within the Federated Edge the device with the "ability to recognize a specific target of an Edge Device" with the higher "quality of information" is then identified. This device can be defined as Edge Master for federation. Edge Master is the device that is presumed to be able to identify and track the target and therefore can also decide if during the movement he loses his ability and can appoint another Edge Master.

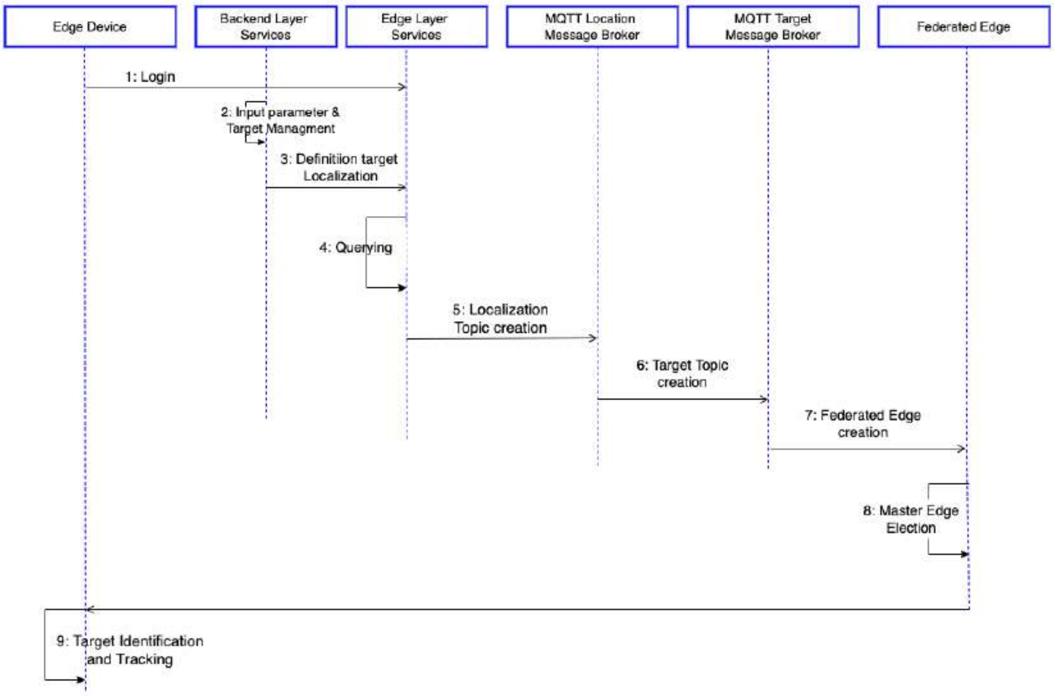

Edge Device
Backend Layer Services
Edge Layer Services
MQTT Location Message Broker
MQTT Target Message Broker
Federated Edge
1: Login
2: Input parameter & Target Managment
3: Definitiion target Localization
4: Querying
5: Localization Topic creation
6: Target Topic creation
7: Federated Edge creation
8: Master Edge Election
9: Target Identification and Tracking

Figure 9.8: Sequence Diagram.

For a better description of the system described, we refer to Figure 11.2. Each device that is part of the described system must be identified through accounting (Step 1 - Figure 11.2). Verification takes place in Edge Layer Services. The input of the target to be searched in the system and of the position takes place in the Backend Layer Services (Step 2 - Figure 11.2). The target model is identified in this component. The identification of the position takes place in the Edge Layer Services (Step 3 - Figure 11.2). This component has the ability to understand which cameras are suitable for target identification (Step 4 - Figure 11.2). At this point, the topics are created in the MQTT Server to put the Edge Devices in contact that are notified of the request. The Edge Devices must first subscribe to the topic related to the position (Step 5 - Figure 11.2) and only after, based on their ability to identify the target, can they subscribe to the topic concerning the target (Step 6 - Figure 11.2). The Federated Edge is defined among the Edge Devices subscribed to the topic of the MQTT Target Message Broker (Step 7 - Figure 11.2). Among the Edge Devices of the Federated Edge, the Master Edge is chosen on the basis of the definition of "ability to recognize a specific target of an Edge Device" with the higher "quality of information" (Step 8 - Figure 11.2). The chosen device will perform the required functions (Step 9 - Figure 11.2).

9.1.5 Final Remarks

This work introduces the concept of Federated Edge defined as a solution to a real problem within Smart Cities. In particular, the system described defines the design principles of an ICT system useful for solving the following problems:

- given an uniquely identified area: understand which camera has the best ability to recognize a specific target;
- given a target: which camera is able to recognize it and at what quality level.

As a next phase of the proposed work, it will be necessary to implement the architecture described and validate the technologies proposed for its operation. For the evaluation of the system, its ability to create a Federated Edge in a time useful for identifying a target will be relevant. In addition, the ability of the devices to be part of multiple Federated Edges in parallel will be considered.

9.2 Optimized Edge-based Federated Learning

Although Cloud Computing has been a great innovation over the years, changing the way computational resources are provided, over time some of its limitations have been molten. Cloud computing in particular has shown weaknesses concerning data security and data privacy. This issue becomes even more important when we think about use cases such as healthcare, which are very common in this paradigm. For this set of reasons, over the past few years, research trends have been pushing the paradigm of Edge Computing, which contrasts precisely with the centralization of the Cloud by trying to overcome its limitations. Edge computing, in fact, moves computation to the *borders*, near where data is generated or collected, i.e. IoT devices. Considered the main purpose of Edge Computing, it makes use of devices with limited resources. Over time, however, advances in technologies are improving these devices, making them increasingly high-performance. This is therefore driving research and new architectures to move computation right into Edge Computing so that all its advantages can be exploited. The main advantage is the security ensured by Edge Computing and by its nature.

For this reason, Google introduced in 2017 an innovative technique that allowed the training of Machine Learning models without compromising its users' data privacy named *Federated Learning* (FL).

In the classical approach, to train a Machine Learning model, the data was shared with a Cloud Infrastructure increasing the risk that the latter could be violated. In this new approach, each Edge Device trains a local model and all the models will be aggregated without sharing data with Cloud. Various advances have characterized federated learning, which, however, still has some open problems. Some of them are caused by the several computational constraints still present in Edge Device.

For this reason, in this work, we propose a new approach to the Federated Learning architecture design in which each client is not necessarily a single Device. Our solution tries to decouple the data analysis flow present in each Federated Learning client, distributing the main tasks across more physical devices. Our solution can overcome the problem of computational constraint in Federated Learning because, the single client is, in our solution a cluster of different physical devices. In this research we propose a new architecture, implement it and we tested it in order to prove the advantages this new approach can bring. The major contributions of this paper are the following:

- we consider the current solution in Federated Learning, especially about Clustering approach;
- we discuss the challenge of Federated Learning current architectures;
- we propose our new approach in Federated Learning client design
- we implement the solution proposed testing it and comparing it with the current, single-node, approach.

We made our experiments for different dataset sizes (from 500 samples up to 10000), considering a FL client composed of two different devices (the *training node* and the *data node*). Results showed that the benefits of our approach increase at the increasing of the dataset size.

9.2.1 Related Work

Background

Federated Learning (FL) is one of the most innovative techniques developed in the last few years. It was designed and created by Google in 2017 in order to perform Machine Learning activities over data without sharing them with a Central Server Infrastructure [270]. Indeed, this innovative approach distributes the training of a Machine Learning model among the Edge Clients got by users themselves. Each user's device trains a partial model that an aggregator entity will aggregate. The first aggregation algorithm introduced was *FedAvg* [271]. The latter exploits the Distributed Selective Stochastic Gradient Descent [272]. This approach aggregates the partial models trained by all devices performing an average of the weights trained to get a global model that is re-transmitted to each client.

According to the first definition of FL, the aggregator entity is a Central Server, often placed in a Cloud Infrastructure. Several research works have decentralized the aggregation of partial models to exploit the advantages of a decentralized architecture [273][274]. This new approach eliminates the central aggregator node envisioned in the first definition of FL in order to avoid a single point of failure, optimize network communications, etc.

Decentralized FL

The decentralization strategy moves the aggregation process from the central server to the edge nodes themselves.

Different architectures and different strategies have been tried to optimize the traditional federated learning architecture distributing the aggregation process according to different patterns [275].

- Hierarchical Architecture. It introduces different layers to decouple the aggregation of the nodes involved by maintaining a central node for the final aggregation [276] [277] [278].

- Regional architecture. It is similar to Hierarchical Architecture but the aggregation is no longer placed in a central node, but it is performed within the edge layer assigned to the respective regions [279] [280].

- Full Decentralized Architecture. This pattern involves the aggregation process in the final edge nodes themselves. Each top layer where aggregation used to occur is now removed and aggregation is implemented in the nodes themselves, which according to different strategies exchange the trained partial model, i.e., the weights [281].

Decentralized architectures, as seen, also delegate to end nodes operations that were previously in charge of the centralized aggregator node, like the aggregation itself. This increases the computational load of the edge node located at the extremes of the architecture.

Clustering

One of the most used approaches in the literature (strictly related to the solution proposed in this paper) was the Clustering of FL clients. This kind of approach tries to aggregate a specific number of clients in order to overcome some well-known problems like the bandwidth usage in the communication of the models and the Non-independently and identically distributed data (Non-IID) [282] [283]. The cluster can be designed according to several criteria. These could be, for example, the location of nodes, their energy impact or the distribution of data [284]. Other works create dynamic clusters in order to improve the accuracy of the model and optimize the aggregation operation [285]. However, all research work on this subject deviates from our solution, which is at a lower level. In contrast to the works cited above, our proposed solution does not consider each physical node necessarily a trainer. As we will see in detail, the cluster we envisioned wants to abstract the federated node itself so as to optimize training, aggregation, and side operations. The starting point of this work is a research that tries to abstracts the Edge Devices according a Edge as a Service paradigm that is considered propaedeutic to Decentralized Federated Learning [286, 287].

9.2.2 Solution Proposed

The architecture proposed consists of a cluster of N nodes representing a single FL client. The solution, therefore, aims to propose a new approach to the design of the main component present in a Federated Learning architecture.

Architecture

The proposed new approach tries to distribute the computational load over several physical devices through the use of precise architecture and precise workflow. Clustering of each FL client allows for exploitation of more computation and allows better management of

collected data and trained local model.
In the cluster, two types of nodes are considered:

- the training nodes that actually train the local model.
- the data nodes for collecting and managing data involved in the training model.

In order to distribute each task inside the devices belonging to the same cluster we packed each function inside a container [288]. In this way, the number of devices for each type is not defined *a priori* but could be managed with respect to the specific use case. The cluster provides more computational resources for each operation.
The training nodes can better exploit the hardware for training operations while the data nodes can only take care of data collection and their pre-processing.

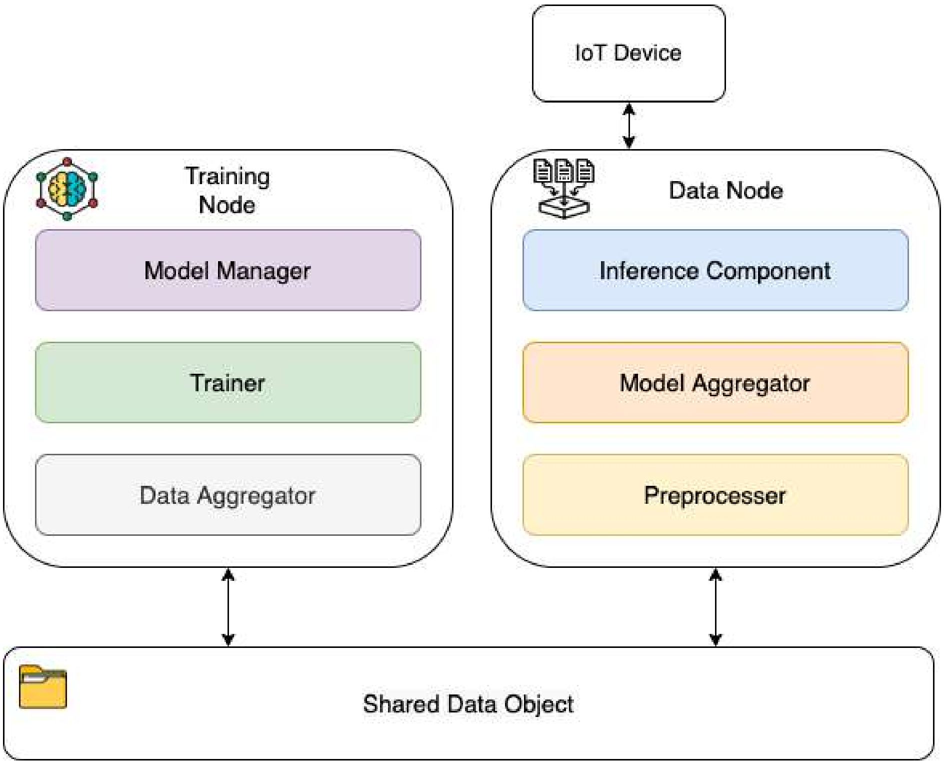


Figure 9.9: FL Architecture with clustered clients

The replica of data nodes, within the cluster, allows for uniform data collected avoiding that training with Non-independently and identically distributed (Non-IID) data could degrade global model accuracy. The Figure 9.9 depicts how the nodes are organized.
Three main components characterize the training node:

- The Trainer is the component that carries out the local model training.

- The Model Manager is the component that manages the local model trained (e.g transmit the locally trained model parameters to the other FL components for the aggregation).
- The Data Aggregator is the component that triggers the training phase when the dataset is prepared. Moreover, it aggregates and uniforms all the data coming from Data Node.

The Data Node, as we said, is in charge to collect data and preprocess them in order to prepare the dataset through which the training phase will be accomplished. As Figure shows, it can be enriched by an IoT Device and it is constituted by three main components:

- The Model Aggregator that carries out the local models' aggregation of other clients.
- The Inference Component that makes the inference with the global aggregated model.
- The Preprocessor manages the collected data in order to prepare them for the training phase.

The Figure 9.9 shows another important component: the Shared Data Object. This represents the entity by which, within the cluster, the Nodes share data with each other. Hence, it is in charge to store the dataset preprocessed by Data Nodes and exploited during the training phase.

Flow

In our solution, as we can see from the proposed architecture we imagine a precise workflow that includes the components described. This workflow generalizes the main steps we have in federated learning. As we have seen, the literature has introduced many algorithms for optimizing the training of the local weights of each device and in their aggregation across the entire federated architecture. Our proposed solution is transparent with respect to these solutions. It investigates at a low level the management of operations leading to the training of the partial model. For this reason, we have defined a specific flow that generalizes them and which, in our solution, we distribute optimally. We can summarize the flow in the following steps:

1. The data collection performed by IoT Layer
2. The preprocessing of collected data for the training phase
3. The aggregation of all data collected
4. The training of a partial model that will be shared among the whole architecture

5. The aggregation of the partial model received by other FL clients

Subsequent to the steps described, it is necessary to include any eventual inference that would occur through the aggregate global model.

The proposed solution decouple these steps so that they can be optimized, parallelized, and distributed over multiple computational resources. The components described are in charge to perform this steps in order to carry out a total local training. To do this, the architecture components described act by implementing the flow through synchronous communications that enable the creation of the local model. The Figure 10.2 shows just how the flow unfolds and how each step is performed by the components described above. The proposed solution therefore not only builds an architecture, but establishes a protocol that orchestrates all the steps within the cluster.

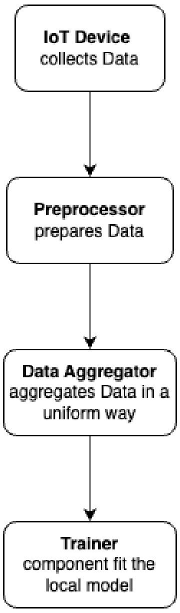


Figure 9.10: Steps involved in a local training

As previously discussed, the work carried out in this paper allows to overcome the unbalanced Non-IID dataset in Federated Learning. Indeed the Figure 10.2 shows us, among the other steps, the aggregation of Collected Data. In this kind of aggregation, all Data collected by different Data Nodes are aggregated in order to prepare a dataset that can be uniform with respect to the other partial dataset of Federated Learning architecture. This kind of approach is analogue to a solution seen in literature [283]. Unlike that solution, the partial dataset is not provided by a Cloud Infrastructure but it’s provided by different Data Nodes of the cluster. The Data Aggregator components can establish how the dataset will be trained in order to uniform the size ensuring homogeneity across the whole FL architecture.

9.2.3 Implementation

In this section, we will see how the described architecture was implemented and what technologies were used. In our work, we carried out a prototype implementation of the above architecture. Specifically, we realized a cluster of edge nodes capable of training a partial model by completing the entire flow starting from data collection and ending with training the model itself.

Infrastructure

The infrastructure in which the implementation was implemented consists of microprocessors, which as is well known, represents the main devices used in Edge Computing. Specifically, we considered in our implementation, a Raspberry Pi 4 with 4 GB of Ram for the implementation of a Data Node and a Jetson Nano (Developer Kit) with 4 GB of RAM and integrated GPU for the Trainer Node.
Both devices were equipped with a Linux distribution, Raspberry OS and Ubuntu 16.04, respectively. The devices equip an arm 64-bit architecture. The cluster implementation was done through the configuration of a Kubernetes cluster. Kubernetes represents one of the major solutions in container orchestration within a device cluster. In particular, k3s Kubernetes distro has been used [289]. k3s was a lightweight Kubernetes distribution developed by the Rancher organization. Due to its lightness, k3s is strongly recommended for Edge Devices. However, it looks like very flexible because it supports all kinds of architecture. In our case, we exploit k3s in order to create an Edge Devices Kubernetes Cluster. Inside our cluster, we have deployed all the components above described. As we have shown in the design section, another important component in the architecture is the Shared Data Object. The latter is the main element by which the cluster can be considered like one single Federated Learning client. Indeed, the Shared Data Object allows the physical nodes to share all the contextual data, like the collected IoT Device Data, the preprocessed data and so on. GlusterFS tool was used to realize that. GlusterFS is a network filesystem that allows the creation of distributed and replicated filesystems among different devices connected with each other [290].

Architecture Components

Architecture Components were each implemented as a microservice within our cluster. In particular, each component was built as Container Image and it was deployed as a Kubernetes Deployment inside our cluster. The components are almost always implemented as HTTP servers because, as we have said, the cluster perform a synchronous flow to train a local model. In particular we have

- The Trainer was implemented using TensorFlow framework and Keras API. The component takes the data from Shared Data Object after they are written from Data Aggregator. It load and trains an existing model saved on Shared Data Object.
- The Model Manager is a Python script that sends the new local model trained to another entity. It can be a cloud aggregator server or, in a decentralized fashion, another Federated Client. It is triggered by Trainer component
- The Data Aggregator is implemented as a infinite loop that check the new data written by Preprocessor component.
- The Model Aggregator is realized as a script that implements the Aggregation Algorithm. This step can be implemented, according to the literature, in several ways but the actual implementation is out of the scope of this paper.
- The Preprocessor Component is the component that preprocesses the data acquired via IoT devices. Even in this case the practical implementation depends by each use-case. We won't deep that.

The inference component was implemented too. It is not present within the whole Flow. It is implemented, even in this case as a HTTP Server. It's not called by internal element but it is external exposed.

Listing 9.1: Training Deployment YAML

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: training-app
  namespace: arch
spec:
  selector:
    matchLabels:
      app: training-app
  template:
    metadata:
      labels:
        app: training-app
    spec:
      containers:
      - name: training-script
        image: alecatalfamo/training:0.1
        imagePullPolicy: Always
        volumeMounts:
        - mountPath: "/home/app/gluster"
          name: task-pv-storage
      nodeSelector:
        device: jetson
      volumes:
      - name: task-pv-storage
        persistentVolumeClaim:
          claimName: gluster-claim-2
```

The Listing 9.1 shows one of the YAML files used to deploy all the architecture inside the Kubernetes cluster. The code shows us, in particular, the deployment of the Training Component inside our Trainer node. As we said, the train component, in our prototype is realized through a Jetson Nano. For this reason we can see that we have specified a particular node for deployment labeled in our cluster with "device=jetson". The other elements of the Listing are the main paramenters foreseen in the Kubernetes Deployment API. Obviously, all the components implemented were developed using Kubernetes and a YAML file. We report exclusively the case of training for simplicity. The structure is the same for other components.

Listing 9.2: Data Aggregator triggering function

```
def watchdog():
    while True:
        if checkAllClassesFiles(N):
            for cla in classes:
                pathWrittien =
                arrayWrittenPaths[cla]

                pathTrain =
                arrayTrainingPaths[cla]

                copyFilesToTrainingFolder
                (pathWrittien, pathTrain)
            try:
                startFit()
            except Exception as ex:
                print(ex)
                print("Fit Problem")
                continue
        else:
            continue
```

Listing 9.2 shows a part of Data Aggregator. In particular, it shows the main functionality of this component. The listing shows part of the implementation of the Data Aggregator component, and in particular the function of *watchdog*. It takes care of calling the Trainer only when in the Shared Data Object, which in our case is represented by a folder replicated via GlusterFS, there is a specified and pre-configured number of files.

9.2.4 Performance Evaluation

The main innovation of this work is the distribution of the main tasks involved in a Federated Learning Client. The main advantage of described solution lies in the distribution of the tasks involved in Federated Learning. Our performance tests will focus on the overhead that can be introduced by the necessary networking. We will also go on to measure the benefits, mainly in terms of timing, that distribution can bring to model training. This in fact represents one of the most onerous tasks for an Edge device that, as mentioned, is usually limited in computational resources. To test the performance of our solution, we considered our prototype implementation, which as we recall consists of a Raspberry PI 4 of 4GB and a Jetson Nano (Developer Kit) equipped with a GPU and with 4GB of RAM. In order to test we have implemented a Neural Network Model for binary recognition. In particular, as we can see in Figure 9.11, we have tested the epoch training time comparing the situation in which all the tasks are distributed in two nodes and the situation in which we have one single node that manages all the tasks. We have tested the training, increasing the number of samples used for the training. In particular, we have compared the situation in which the deployment is carried out in a single node and the situation in which these are distributed across two nodes: The Data Node and the Trainer Node. As we can see the result obtained is an increase of the training time for the single node context. Obviously, this comparison can appear not

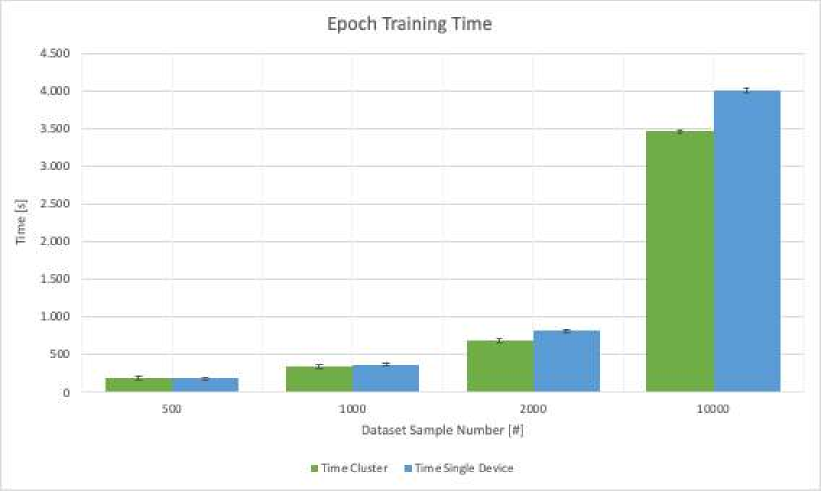


Figure 9.11: FL Architecture with clustered clients.

meaningful, but we have to say that the trend can improve if we increase the node in the cluster. We have tested a cluster of two nodes. We can imagine implementing, as said, even clusters with a greater number of nodes.

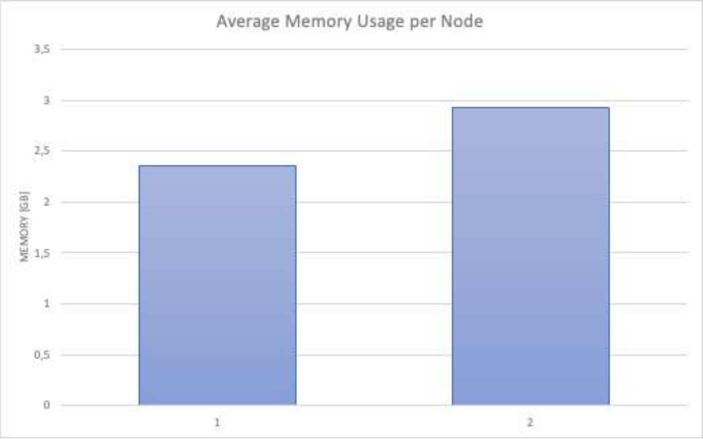


Figure 9.12: Average Used Memory per node in Training phase.

Figure 9.12 shows what is the average memory per node, used during the training of the local model. In particular, we can see that the average of used memory per node, is lower in our solution in which all the flow is managed by a cluster. In our node indeed we have more processes that have to manage all the flow and the cluster but the average used memory in a node is still less respect to the solution in which we have a single node.

The Figure 9.13 expresses a similar concept. If we consider, as a percentage, the memory used compared to the total memory available, we can see that the tests confirmed what was already guessed. Our solution provides a greater quantity of computational resources for the management of the whole flow.

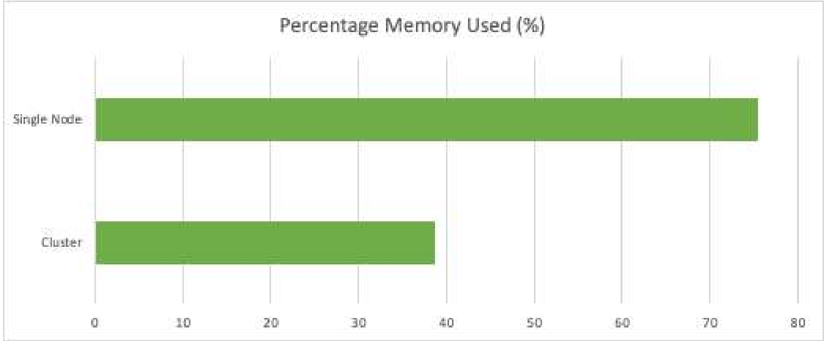


Figure 9.13: Percentage of memory used to total.

9.2.5 Final Remarks

The most challenging issues in Federated Learning are strictly related to the Edge Computing devices and them nature.

In this work, we try to give a solution for some of them. The research proposes an innovative architecture in which the concept of Federated Learning Client is abstracted. With our solution, the client of Federated Learning architecture is decoupled and all the tasks involved are distributed in a smart way. This solution can bring more resource computational and can overcome the unbalanced distributed dataset issue that is frequent in Federated Learning and that can deteriorate the final accuracy of global model. The test performed in our work prove the positive trend in time response about the local model training.

In future works we plan to perform other tests about our solution, in particular, about the computational resource spent in our solution. Moreover, we wants to analyze the energy impact that our solution bring. In next works we want also to study the possibility to create a dynamic cluster FL architecture in which the clusters client are not established a priori but they can dynamically change in order to improve specific criteria of the model (accuracy, time, and so on).

CHAPTER 10

Innovative Technologies in Smart Mobility Applications

To spread the concept of sustainable mobility it is necessary to encourage citizens to use zero-impact vehicles instead of private cars. An example of this practice comes from a partnership between the University of Messina and the Municipality of the City of Messina (Italy). Partnership has made it possible to develop a digital application to assign electric bicycles to citizens, free of charge for a limited period of time. The key issue addressed in the development of such an application is security, both in terms of secure authentication of citizens who access the service which tracks the entire process of assessing the application and assigning the bicycle to the user until it is returned. Another aspect addressed concerns the use of technologies such as Big Data, Cloud, Edge Computing, and IoT analysis. Through these technologies, it is possible to design both decision support services and the management of smart cities and services for users. The work described in this chapter presents an innovative device for the safety of cyclists in urban areas. The device is designed to ensure continuous connectivity. Connectivity it can be global (via the internet) or local thanks to the use of the file mesh net. Furthermore, the designed infrastructure provides cyclists with real-time information on the urban area in which they find themselves crossing, starting from the data collected by the smart city. During the implementation phase, it was of particular interest to test the network operation and energy performance of the device in the field. This chapter reports the design approach adopted and some results on the efficiency of the services.

10.1 Secure Access to Moblity Service in Smart City

Smart Cities have become a concrete reality all around the world. They enable innovative services for citizens to improve their quality of life, but also allow public administrations and private stakeholders to improve their processes and increase their business. Although the advantages in promoting Smart City services and applications are evident, their actual implementation is limited by security issues [291]. In the last years, some researches tried to improve the security for Smart City digital environments. In particular, a huge number of Information and Communication Technology (ICT) platforms and tools were born to manage all main aspects of a Smart Cities like Access Control Management (ACM) and Authentication, Authorization, Accounting (AAA) policies. However, investigated results are hardly to be coupled with efficient or worthy applications, especially whenever these involve public administration and e-government processes [292].

In this section, we try to overcome these issues presenting a new digital service that aims at pushing sustainable transportation in a Smart City by using electric bicycles instead of private and public vehicles powered by carbon-based fuel. Through a partnership with the Municipality of the Messina city (Italy), we developed a solution able to assign citizens electric bicycles, free of charge for a limited period of time. In particular, we implemented our solution addressing security issues both in terms of physical recognition of the user that accesses the service and for tracking of the whole assignment process, from the request of the user to have a bicycle to its restitution. To achieve such a goal, we adopted the 2FA (two-factor authentication) and the Blockchain technology integrated in the municipality processes for bicycle management. The Blockchain component ensures the integrity and the secure tracking of each step involved in the process [293], [294]. Specifically, this section describe the design of the web platform to instantiate the services for requesting electric bike (e-bike), and for their assignment and access management. Furthermore, we provide implementation details, and some experimental results on the effective adoption of the developed solution in the Messina city use case.

10.1.1 Related Work

In literature, some works address AAA issues in Smart Cities. In [295], the authors propose a new architecture for Smart Cities, also comparing several platforms and approaches that already exist. In the proposed solution, the Cloud platform Smart City oriented FIWARE is enriched by the Blockchain technology that is exploited in two cases. In the first one it is used to replace the database that collects all the data related to registered users and in the second case it is instead used to store the access policies, in a eXtensible Access Control Markup Language (XACML) based architecture. This work is interesting because it creates a decentralized architecture taking the advantages of Blockchain. Nevertheless, the security for the end user is not really increased. In fact, the authentication managed through the proposed approach does not include multi-factor authentication. The architecture described can be exposed to well-known weaknesses of single-factor authentication like the use of weak and reused passwords by users. An interesting work based on an e-Health application for Smart City is proposed in [296]. The research improves the already existing OAuth2 protocol to avoid any disclosure of users credentials and user's sensitive information. The proposed architecture avoids that any user information is stored in mobile application or in browser. The solution is powerful because it takes the advantages of Elliptic Curve Cryptography (ECC) based signature to create a custom flow ensuring the privacy of end user. Furthermore, the designed flow is perfectly integrated with the well-known OAuth2 standard. Unlike the use case proposed in this work, this solution is also subject to threats related to the non-use of any form of multi-factor authentication. For this reason, a weak password, or a reused password could represent a threat. The designs an anonymous authentication approach for Smart Cities applications and mobile Cloud computing environment is presented in [297]. The application adopts a Trusted Third Party (TTP) to perform the authentication of both applications and users in an asynchronous fashion. Even in this work the solution does not include a multi-factor authentication. Furthermore, the presence of the TTP makes the logic business linked to a highly centralized authentication. If an hypothetical attacker violated the TTP, he/she could impersonate a registered user or applications registered. On the contrary, our solution is microservices based and decentralized. Furthermore, our solution implements a two-factor authentication avoiding any single point of failure in user's identity management. The research described in [298] an authentication service provided by the Agricultural University of Athens (AUA) to the Erasmus exchange students in order to be compatible with the European eIDAS regulation. Even this application presents some weaknesses. In fact, the credentials are released by the AUA and this could represent source of internal threat. Even in this case the authentication phase is carried out only by simple credentials including username and password. Advanced Internet of Things (A-IoT) Multi-Factor authentication mechanisms are discussed in [299]. Specifically, it is highlighted that for A-IoT services in a Smart City environment, the single factor authentication is not enough. Although this work is interesting and we agree with the assumptions made by the authors, it is quite theoretical. On the contrary, in this work, we present a concrete Blockchain based 2FA solution adopted in a real service deployed in the Messina city.

10.1.2 MuoviMe Services Design

Now we present the design of MuoviME application for the long-time renting of electrical bikes to citizens in a Smart City. MuoviME comes from the requirements of the Italian project "Messina - A scuola e al lavoro con il Trasporto Pubblico Locale. Iniziative per promuovere la mobilita' sostenibile" that aims at promoting sustainable mobility in the city of Messina (Italy). The project includes among partners the University Messina and the Municipality of Messina. In particular, the project has the main purpose of discouraging the use of carbon-fuel powered vehicles and promoting activities that increase sustainable mobility within the city.

To achieve such a goal, we thought to a public web platform where each citizen can submit a request to have, if available, an electrical bike assigned for free and for a limited period (e.g., max 90 days). Thus, we designed such a platform considering that the efficiency and the success of the offered service strongly depend on the overcoming of several security issues. Specifically, the platform must always be available and tolerant the failures that are typical of a centralized system, thence avoiding any possible bottleneck. Furthermore, the platform has to implement a strong authentication system able to guarantee the association between the digital identity of a citizen and the physical access to the assigned bike from the same citizen. In fact, at the time of delivery of the bike, the user, who will get it, must be authenticated, authorized, and recognized with the digital account that made the request. Moreover, the platform must guarantee that each phase of the bike rental process, is tracked by a third distributed system in order to ensure the whole process integrity. In our case, such a system is represented by Blockchain. In the following, we present the reference architecture we designed for the whole bike rental system workflow.

Architecture

The management of electric bikes in the MuoviMe application has been designed considering the benefits of a microservices-based approach [300]. In particular, the whole solution has been organized as a set of independent microservice components that interact each other and that can be deployed and dynamically managed taking the advantage virtualization in terms of resource optimization. Moreover, the microservices approach allows us to avoid the presence of a Single Point of Failure in the system, increasing the fault-tolerance of the service and its fast restore in case of attacks or technical issue [301].

The main microservice components of the MuoviMe architecture are shown in Figure 10.1. They include:

- The Web Server, that is globally reachable, through which each user can authenticate itself and access the MuoviME application.
- The OTP Management Module for the management of the 2FA needed by the user for the renting of the e-bike.
- The Blockchain Gateway allowing the communication with the Blockchain. It can abstract different kinds of Blockchain, i.e., either private or public.
- The Database Module ensures persistence data of Web Server and it interacts with OTP Management Module for the generation of OTP.

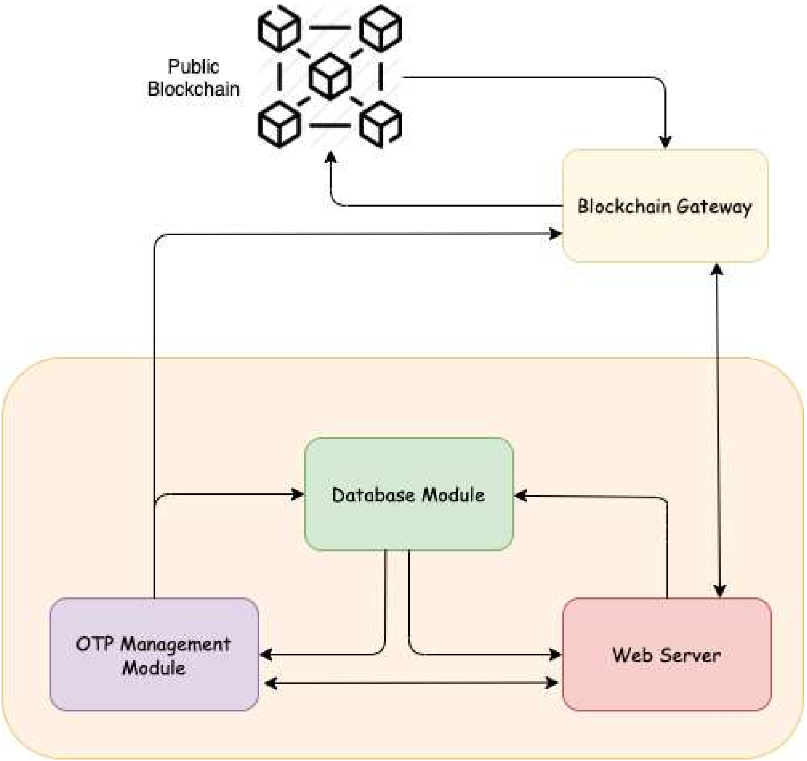


Figure 10.1: Reference architecture.

Workflow

The actors involved in the MuoviME application are:

- The End User. The person who makes a new e-bike rental request. He/she interacts with the system in order to make a new request. This request is authenticated through the provided authentication platform.

- The Officer. He/she has the responsibility to validate or invalidate the e-bike rental request performed by end user. Only if the Officer validates the request, the MuoviMe workflow can go ahead.

Before the start the e-bike rental workflow, an initialization phase is required. Indeed, in this phase, the officer generates the key-pair useful for signing the validation or invalidation of each request. The generation has to be carried out only with the presence of the platform administrator. Indeed, once generated the key, the administrator signs the key generated in order to certify it. The workflow, from the initial request of the end user to the delivery of the bike is described in Figure 10.2 and consists of the following phases:

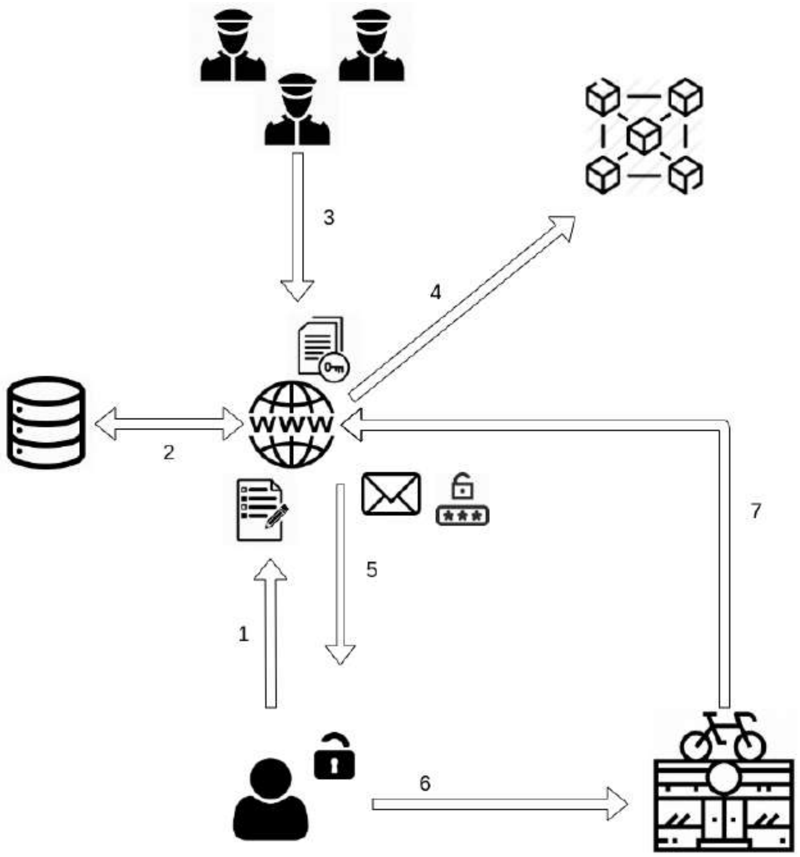


Figure 10.2: Use-case flow.

1. The End User makes a new request through the Web Platform. The request involves personal data of the User, identity document and so on.

2. The Web Server stores the request involving the end-user information inside Database.

3. The Officier checks the request submitted by the end-user and all the pieces of involved information: if the request is valid, he/she can validate the request. This step is not synchronous. The Officier can choice to validate more than one request in a single session. The request validation is signed with the private key generated in the initialization phase.

4. Once the request is validated, the platform, through the Blockchain Gateway (Figure 10.1), writes the information contained in the request in a Public Blockchain. In this phase, the platform makes a request to the OTP module requiring a new OTP for the delivery of the requested e-bike. The OTP generation is realised according to an innovative approach [302]. Indeed, the OTP module (Figure 10.1) is deployed in a distributed fashion and it interacts with the Blockchain in the generation process. The OTP module is organized in microservices allowing to not be subject to Man in the Middle or DDoS attacks. Furthermore, the OTP module uses the Blockchain to store the seed used in the generation of the OTP for the user.

5. The End User receives the OTP through which he/she can collect his/her e-bike at the designated collection point.

6. The end user goes to the collection point to collect the requested e-bike. The system after the evaluation of OTP, through the exposed Web Server, records the delivery and all the information related to it.

Once the period of use is completed, the end-user has to give back the e-bicycle. In particular, he/she has to bring back it to the recollect point in order to make available again the bicycle used to new users.

10.1.3 Implementation

Microservices and web platform

As we said the architecture is focused and based on microservices. The implementation strategy is based on the Linux container concept [303]. In particular, the implementation of containers for each module considered in the architecture took place through the Docker container engine. Docker is a powerful tool widely supported by a community of developers. Indeed, the official registry containing all the images, always knows as "Docker hub", provides a huge number of images useful to deploy a various of services. In our case through Docker we have deployed the Web Server and the Database module. In particular, to develop

the Web Application, we have used the Content Mangement System Drupal.
Drupal is characterized by a strong modulation. Moreover, it allows integrating custom modules, using PHP language, for the implementation of advanced functionalities. The Database module was implemented and deployed through the MySQL 5 Docker image. As said in the design section the Database module supports both Web Platform (e.g, Drupal Instance) and OTP Module.

End-User Functionalities

The main functionalities for end user were implemented through the available Drupal modules, whereas advanced functionalities were developed through an ad hoc custom module. For example, the Principal component of User Interface is provided by Drupal and by external modules supported by the community. Authentication and Authorization features are accomplished by native module of Drupal (core). The authentication is implemented through well-known and classical techniques. In particular, it could be implemented through a Lightweight Directory Access Protocol or relational database. Other important graphic functionalities were developed in Drupal by well-known external modules (e.g., web form). As said for the advanced safe functionalities a custom ad hoc module was developed. The latter implements several functionalities. First of all, the it integrates new tables in the Drupal default database that store important information about request, user and all workflow phases.

The integrated tables are three:

- *Submitted Requests* is the table that stores all pieces of request information useful for the validation and delivery of the e-bike.
- *Sellers Updates* keeps track of all digital warehouse.
- *Officier* stores the key-pairs of Officiers.

The first table is populated when the web form for the new request is submitted by an end-user. This mechanism is possible thanks to a powerful tool provided by Drupal, i.e., hook. This is an interface that allows a custom module to reuse both data and services provided by other installed modules. In this case a hook intercepts the submission web form and stores in the table some pieces of information related to the *Submitted Requests*. The second table is updated with the same mechanism. A specific web form is provided for the seller and each submission inserts a new row in the table *Sellers Updates*. The table is updated when a disclaimer delivering email has been send decreasing the number of submitted bikes. Another important hook that is used in our custom module is the "hook menu" of module *menu*. This hook allows creating a new path of the web site through a callback function that runs when the path is called in the browser. This hook allows integrating in the custom module several Representational State Transfer (REST) Application Program Interfaces (APIs) that can be called from the front-end. Moreover, the modules allow integrating javascript code. As we will discuss in the following, javascript code enables Asynchronous JavaScript And XML (AJAX) calls allowing key-pair generation and validation.

KeyPair Generation

The KayPair Generation is implemented client-side through a javascript code embedded in our custom Drupal module using the AJAX approach. In particular the javascript PGP library is used to implement the geneartion of KeyPair with Elliptic Cryptography. The curve "ed25519" is used for the key-pair generation. The private key generated is ciphered with the Advanced Encryption Standard (AES) algorithm, using a password got by user-interface. After the creation and the encryption of private key an AJAX call communicates with the "officier/generation" endpoint exposed in the custom module developed through the "hook menu". The back-end stores the private key encrypted in the *Officier* Table.

Validation

The validation execution is the same of the key generation one. We have a javascript code that carries out a REST call thorugh AJAX, in an endpoint created with the "hook menu". The analogue callback function registers the validation or the invalidation storing the related data in the BlockChain transaction. In the validation the officier inserts through the user interface the password of KeyPair previously generated and signed by administrator, and decrypts the Private Key previously got by back-end and stored inside the database. With the private key the officier signs a JavaScript Object Notation (JSON) document in which the results of validation and the possible motivation are contained. The signature is stored in database through another exposed REST API.

Blockchain Communication

The Blockchain Gateway (Figure 10.1) is a container that acts as a proxy for communication with the adopted Blockchain system (either private or public). It is implemented through a python server built through the flask and guncorn frameworks and containerized, once again, through the Docker system. In particular for the implementation of private Blockchain Hyperledger Fabric was used whereas for the public Blockchain Ethereum Blockchain was considered. The Public Blockchain implementation leverages the Infura web-based Application Program Interface (API) [304] to interact with Ethereum. The Smart Contract is written in JavaScript (in the case of Hyperledger Fabric) and Solidity (in the case of Ethereum) programming languages. The Blockchain Gateway ensures integrity of Officier decision through a Smart Contract that is structured to store hash digest of the officier id and the information about request validation.

10.1.4 Experiments

In the MuoviMe application, the Blockchain system could represent an overhead source that has to be carefully assessed since it plays an important role in the OTP generation process. In fact the response time introduced by Blockchain in the OTP Generation could negatively affect the user experience on the whole system. For this reason, experiments focused on the OTP Generation scaling up the system from the point of view of number of users who simultaneous send e-bike renting requests to the MuoviMe application. Specifically, we assessed the mean response time of the OTP module that, as explained, interacts with Blockchain system for the delivery of OTP. In particular, we compared the use of both private and public Blockchain systems. The difference in execution time expressed in seconds between the two system implementations for generating the OTP is shown in Figure 10.3. The two approaches for OTP generation are shown on the x-axis. The processing time, expressed in seconds, is reported on the y-axis. We can observe that the public Blockchain approach takes five times longer to generate the OTP. This is due to the time required to mine the seed and obtain it from the public Blockchain system. Instead, both solutions take about the same amount of time to alert the freshly created OTP.

10.1.5 Final Remarks

In this section, MuoviME, a solution to promote sustainable mobility in the city of Messina (Italy), was presented, highlighting techniques for the physical recognition of users in the

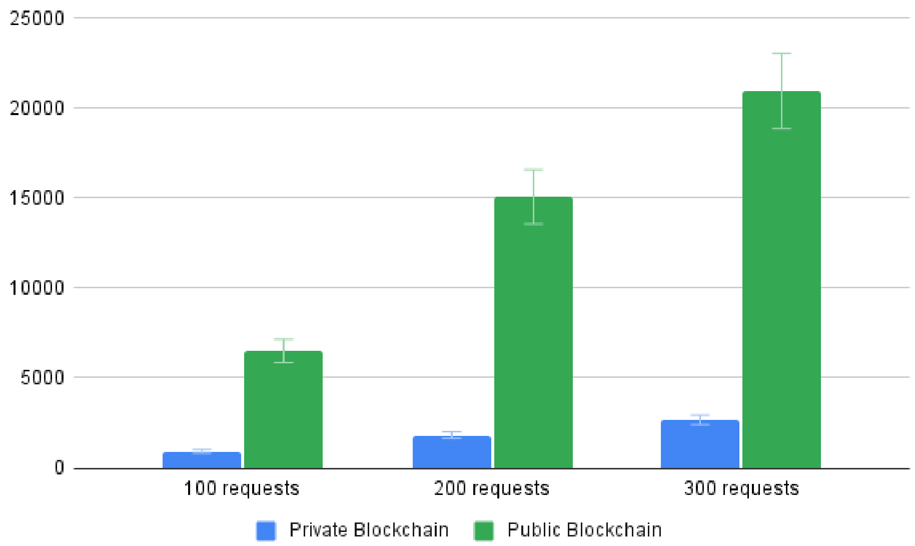


Figure 10.3: OTP time response.

Smart City context. With our solution, the state of the art regarding access control by users in a Smart City has been advanced. Going into detail, the project, carried out in collaboration between the University of Messina and the the Municipality of Messina distributed several e-bikes to citizens promoting and encuraging concrete sustainable mobility policies. Specifically, we developed a web platform that that carried out the whole workflow of e-bike request and assignment in a totally digital fashion, exploiting advanced encryption and Blockchain technologies to guarantee strong authentication and integrity, over the time, of the information entered during the request and during approval. The Use-case described has brought concrete results. Environmental benefits was proven starting from the distance travelled by users.

As expected the use of a public Blockchain increases the overhead in the workflow and in the OTP generation. Anyway, the Public Blockchain represents a strong distributed solution and waiting for a minute could be consider acceptable.

10.2 Edge System for Users in Smart Environments

Today's cities are facing a revolutionary era in urban mobility due to several factors, such as their continuous growth and the concentration of human activities. To prevent and solve mobility-related problems, such as traffic congestion, air pollution, and their potential link with health risks, cities are investigating new solutions for improving urban mobility to meet the actual demand of citizens living or moving around the cities every day. The European Union encourages projects in the field of Smart Mobility and Sustainable Mobility [305]. One of these is the URBANITE project [306], funded under the EU-H2020 funding program. One of the main objectives of URBANITE is to promote the use of disruptive technologies in emerging Smart Cities, for example through the use and analysis of Big Data, AI (Artificial Intelligence) algorithms, etc. The aim of the project is to provide end users with a series of smart technological tools able to support innovative digital transformation in urban mobility management. Bilbao, Helsinki and Messina. The main challenge of the city of Messina for the next few years is twofold: on the one hand, it wants to build new mobility services able to satisfy the needs of citizens, and visitors, allowing them to move and cross the city with continuous services; on the other hand, the challenge consists in optimizing the management and interaction between the different mobility and monitoring systems, and services available in the urban area of the city of Messina, reducing waste of resources and costs for the Public Administration. Starting from the Messina use case, this work proposes the design and prototype of a device that can be usefully adopted to provide value-added urban mobility services within the city.

With this work, we want to propose a device that increases the safety of bicycle users in urban areas. The prototyped device must be connected to an electric bike and continuously powered. The Edge Device is able to collect data on the user's cycling performance and share it with other cyclists through a designed mesh network. In this way, cyclists provide information that enriches the dataset of data collected by the sensors in the smart city, which are subsequently processed within the central city cloud. The increase in safety for the cyclist depends on the ability of the device to provide the user with information about the surrounding environment, to allow him to have a complete overview of what is happening in terms of mobility. The problem addressed focuses on the ability of a device to acquire information from the device, process it, and provide it to the user. Furthermore, the ability of the device to operate for an adequate time without the need for main power is evaluated to assess whether the system is usable during urban travel.

10.2.1 Related Works

Nowadays, smart cities are moving towards new perspectives of sustainable mobility adopting smart models that see the bicycle as the most used transportation by citizens. Electric and non-electric bicycles are used both for normal daily commuting and sporting activities. Furthermore, moving around with a bicycle daily can lead to positive results on health. In [307] a system is proposed that uses smartphone sensors and additional devices to collect information on users' cycling routines. In particular, through the accelerometer of the mobile phone, it is possible to detect the falls of cyclists also providing necessary support in case of accidents. In [308] the authors present a smart bike with a monitoring system for cyclists via the IoT (Internet of Things). The system allows real-time monitoring of the cyclist in terms of health and performance conditions. Monitoring is done using heart rate, pulse oximetry, magnetic reed sensors, and GPS (Global Positioning System) modules. The sensors are connected to the microcontroller, to the Wi-Fi module, and, the data is accessible via an IoT platform or application. The work presented included the mounting of sensors on the bike while others monitor the cyclist. The application interface displays parameters such as heart rate, pulse oximetry, speed, distance traveled, and position for the cyclist and trainer to monitor their health and that of the cyclist during training or tournament. The Connected Bike project is presented in [309]. Even in this work, the authors use a variety of technologies, both hardware, and software, to provide cycling enthusiasts with a modern alternative training solution. A trainer can monitor online, through a web application, some of the important parameters for training, in particular the speed, cadence, and power generated by the cyclist. In addition, the trainer can see the cyclist's position with the help of a GPS module. To acquire data with a low percentage of error, the devices inserted on the bicycles must be equipped with calibrated sensors. Sensor calibration is a very delicate subject and subject to numerous studies. In particular, in [6, 12] the effects of aging on the sensors are analyzed and an on-the-edge self-calibration system is implemented. Furthermore, in the literature, there are different IoT-based solutions for sports analysis, which aim to improve performance, coaching, and strategic information. The authors in [310] present a report on the experience of using cutting-edge IoT technologies in cycling. In this study a group of cyclists can form a reliable and energy-efficient mesh network to collect and process real-time sensor data, such as heart rate, speed, and location. The collected data is analyzed in real-time to estimate the performance of each cyclist and obtain immediate feedback. The information drawn from the tracking of journeys with cycles can also provide information on how the vehicle is used [311]. In general, commercial vendors provide large volumes of data unspecified, but their products are used primarily for motorized travel or are in the early stages of development. Readily available data sources and their applications are more focused on modality-specified data, which has enabled several non-motorized travel studies, including travel pattern identification, route choice modeling, exposure to accidents, air pollution, and evaluating the provision of new facilities - but mainly focus on cycling. In their study, the authors focus on issues related to privacy and the accuracy of the data collected. In this case, it is important the access control and the security of the data [2]. In [42] the results obtained from a Sustainable Mobility project in Messina are described. The application presented aims to encourage citizens to use low-impact vehicles instead of private cars. Through a partnership between different stakeholders, a digital application was developed to assign electric bicycles to citizens, free of charge for a limited period. The authors describe the IT security problems, both in terms of secure authentication for citizens who access the service, and in terms of traceability of the entire assignment process. The flow is described from the user's request to the return of the e-bike. The solution adopted uses two-factor authentication (2FA) and Blockchain as the main technologies being implemented. Innovative and advanced smart devices and virtual devices are described in [43]. Authors have designed, for one use case in the city of Messina, an abstracted component characterized by specific high-level functionalities. The system offers the chance to access the needed information with the most appropriate frequency and accuracy, avoiding information overload and allowing a more efficient computation. In [148] authors show the use of customized generic Edge devices to carry out multiple activities at the same time, also focusing on how the proposed solution can lighten the work of cloud infrastructures. The presented concepts were implemented and tested in a real use case in the city of Messina by means Function as a Service (FaaS) paradigm. The proposed work allows users to perform multiple tasks on the same device. From the studies carried out, it emerges that monitoring the performance of cyclists is a very important issue. Several studies have been carried out for data acquisition, monitoring leaded to increte results. From the state of the art, it emerged that data on mobility are collected in various different cities, and, also in Messina (use case of the URBANITE project). The proposed work aims to combine the data collected by a specific device and the data available on the data lake in the city to create an always-connected information system for bicycle users in urban areas. The designed device allows cyclists in the same geographical area to have always updated information that takes into account the situation in real-time in the city. Finally, the proposed work aims to demonstrate that the simultaneous use of low-cost sensors allows realizing an innovative device that allows to carry out the required switching.

10.2.2 Motivation

Public administrations, as well as companies, are trying to study the traffic flows that move in the smart city. The interest in this issue is given by the fact that more and more people move by different means, and therefore it is necessary to guarantee the control of greenhouse gas emissions but at the same time efficient services. Bicycles are becoming increasingly important in urban mobility. These are vehicles that can move even with zero emissions without causing large traffic flows. In many cities, however, the culture of using the bicycle has not yet been consolidated, this is one of the factors that cause danger for those who use it. Moreover, to encourage the population to use the bike it is necessary to guarantee the safety of this transportation through a new concept of community that makes the vehicle always connected and reachable.

Figure 10.4: Reference Scenario.

The idea behind this paper is to design and prototype a software system consisting of Edge devices and a cloud system. Edge devices must collect data to be sent to the cloud and ensure a continuous connection with other devices of the same "family" geographically close to each other (Figure 10.4). However, the device is based on a cloud that collects and processes data to be provided to users of the system or to support the AI algorithms. Furthermore, the system proposed aims to create a sort of social network between users of green vehicles in urban areas. Thanks to the prototype presented in this work, it will be possible to share routes, alarm events in the case of illness or accident, and share dangers on the road or points of interest in real-time. Our challenge is to design a system that is hybrid concerning connection technologies and therefore is always connected with nearby devices via a mesh network, while connected to the world via a 4g/g5 connection and sends data to the cloud. Furthermore, the impact that this type of technology has on energy consumption will be investigated in order to verify its feasibility.

10.2.3 Design

The high-level reference architecture of the proposed solution is shown in Fig. 10.5. The System Architecture is composed of 4 main levels and different components that communicate with each other through REST API:

- Sensing Layer: is the data acquisition layer, and it is composed of 2 components:

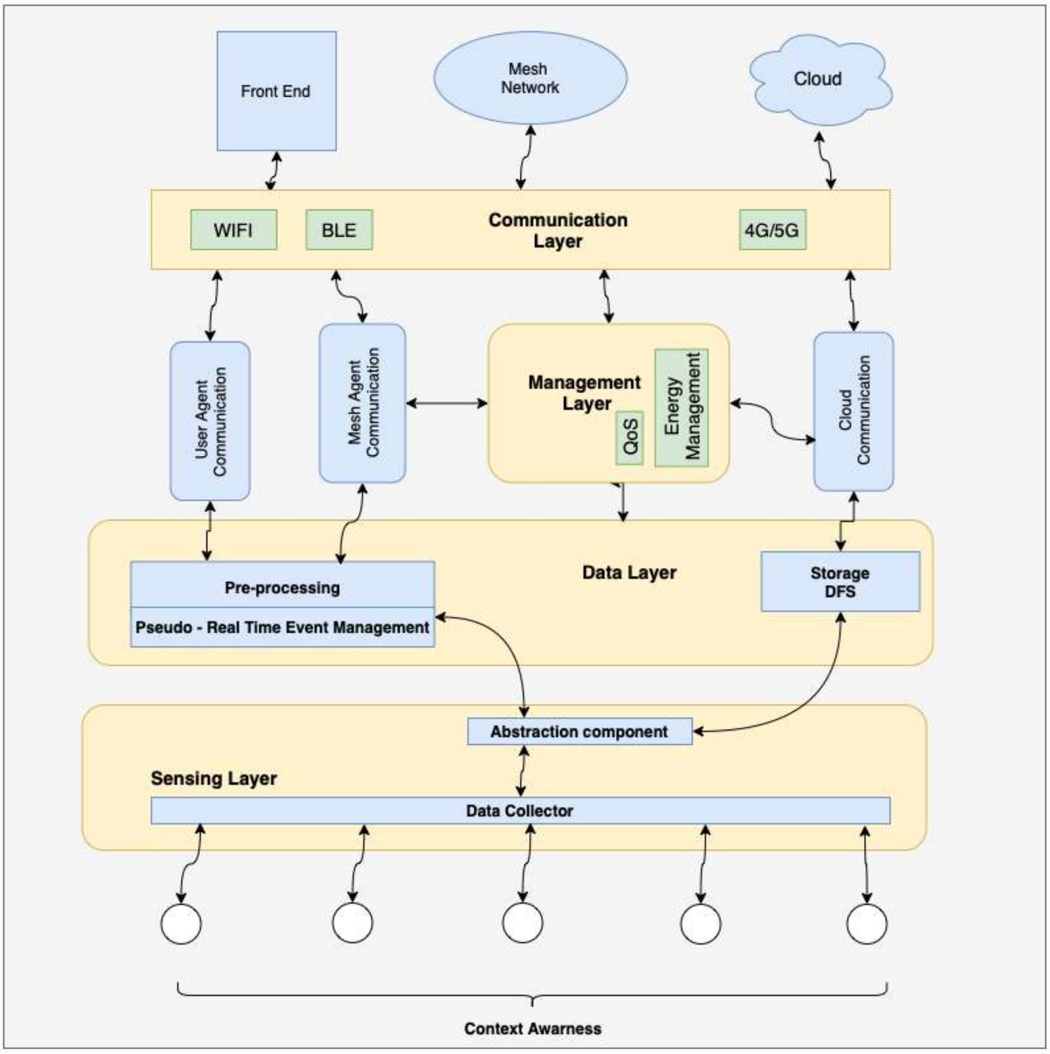


Figure 10.5: General Architecture.

  - Data Collector: this component acquires sensor data. It is optimized for the specific sensor that sends data;
  - Abstraction layer: prepares the data for the top layer. This component represents the interface with the Data Layer;

- Data Layer: it defines the storage and pre-analysis capabilities of the data before it is used. This layer consists of:

  - Storage DFS: it is a distributed file system that maintains the data collected by the Sensing Layer. This component mainly deals with providing data to the Cloud Communication component which is an agent that transfers data to the Cloud through the Connection Layer in presence of an available connection;
  - Pseudo - Real-Time Event Management: this component is composed of an agent that based on the acquired data can trigger an event (i.e. turn on a led, send an

alarm) or perform a pre-processing on the data. Data processing is required to interact with the front-end, and the system user via the User-Agent Communication or with the mesh network via the Mesh Agent Communication. The Mesh Agent Communication through the Communication Layer allows communication between the single Edge Device and the Mesh Network.

- Management Layer: is the architectural core component. This layer aims to guarantee high-quality service in terms of accuracy of the information provided and must ensure efficient management of the energy used by the Edge device. Management Layer coordinates all data acquisitions and their processing as well as monitoring communications and all energy consumption concerning every single operation. In this layer algorithms contribute to the optimization of processes;
- Communication Layer: It defines the communication protocols between the device and the outside. Depending on the recipient of the message, it uses BLE (Bluetooth Low Energy) or WIFI (Wireless Fidelity) protocol or even both for connection with a mesh network. The connection with the cloud is managed by 4G/5G technology.

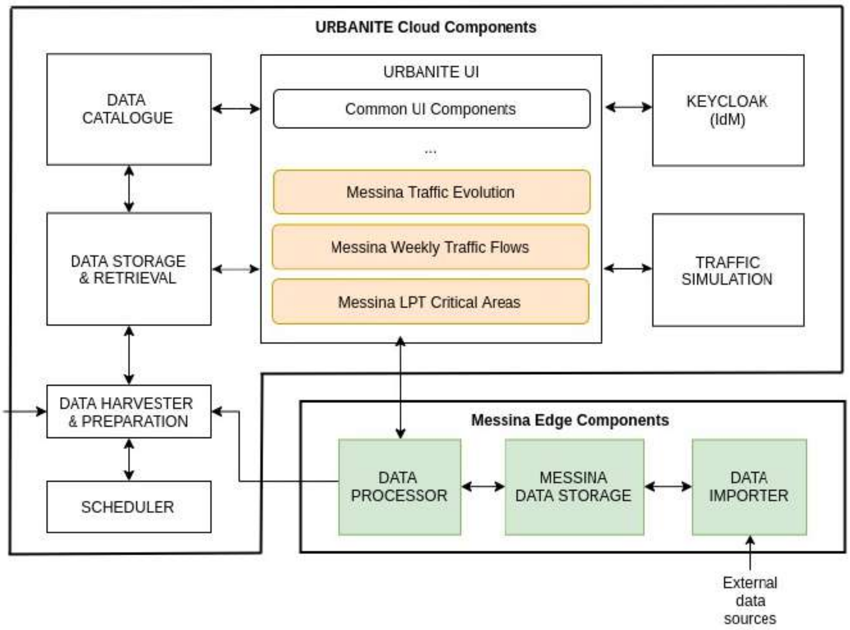


Figure 10.6: Messina-URBANITE Architecture.

In the use case of Messina, the Cloud component in the above architecture has been enriched with new dedicated components at the Edge level, which fully integrate the existing Cloud ecosystem, as shown in Figure 10.6. In particular, a local component called *Messina Data Storage* has been added. This component acts as a support for the parent component *Data*

Storage & Retrieval (reported in URBANITE Cloud Components) through the *Data Harvester & Preparation* and is filled with data by the *Data Importer*. The *Data Processor* allows both to expose the data via Restful API and to process them ensuring correct formatting. Finally, within the *Urbanite UI*, three new specific components for the Messina use case have been built: *Messina Traffic Evolution*, *Messina Traffic Flows*, *Messina LPT Critical Areas*. As well as available for URBANITE UI, the REST APIs are made available to interface with the Level 2 proposed in the general architecture.

Figure 10.7 describes the high-level architecture of the Edge device.

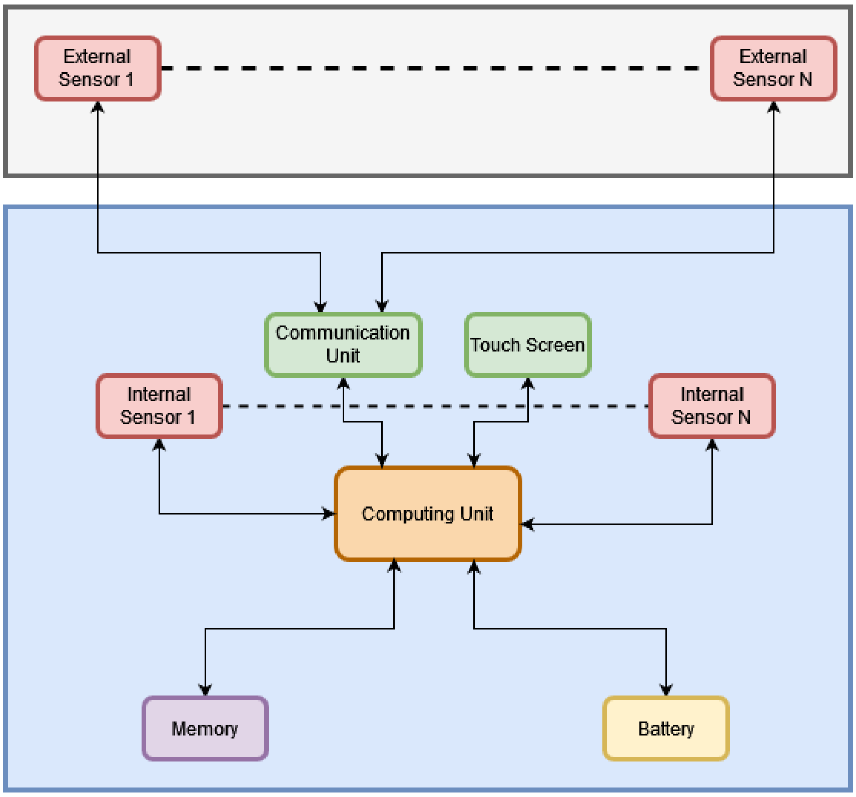


Figure 10.7: Edge Device Architecture.

The Edge device is composed of several components that interact with each other:

- Computing Unit: it constitutes the core of the device. It contains the computing unit and all the components necessary for interacting with the external components;
- Battery: it is an external rechargeable device that powers the system;
- Memory: it allows data to be stored permanently;
- Communication unit: it is a unit used for communication between the computing unit and the sensors external to the device. The communication can be BLE, WI-FI, etc;

- Internal sensor: they are sensors that allow the detection of data and are located inside the device. Examples are the accelerometer for detecting acceleration, and the GPS receiver for locating the bike. The system can also integrate sensors for medical use to evaluate the cyclist's physical performance;

- External Sensor: these are sensors that can be placed on the bicycle or on the cyclist to allow the acquisition of specific data. One example is a reed sensor needed to evaluate the pedaling cadence and the wheel rotation frequency;

- Touch screen: it allows the user to interact with the Edge Device.

10.2.4 Overview Reference Technologies

In this section, we will describe the hardware and software technologies used for the realization of a prototype of the Edge device. The presented device allows users to have information on the instant speed of the bike, travel distance, slope of the road, inclination of the bicycle, and ambient temperature. In addition, sensors will be used to calculate the pedaling cadence

Hardware technologies

- Raspberry Pi W Zero: it is a single board created in 2017 by the Raspberry Pi Foundation and represents the Computing Unit of the Edge device. Energy consumption is the particular strength of this model. It is estimated that in cases of maximum use of resources, consumption is around 1W. This feature makes it easy to use the device even with a battery. Furthermore, the small dimensions (65mm × 30mm × 5mm) make it suitable for the required use. As for connectivity, the card is equipped with an 802.11n 2.4 GHz wireless LAN connection and a Bluetooth 4.1 Low Energy connection. The slot for the micro SD comes used for both loading the operating system and storing data. The Raspberry Pi W Zero is equipped with a Broadcom BCM2835 system-on-chip, with an ARM11 microprocessor running at 1GHz and has 512 MB of ram. It is possible to connect other components to the board through the GPIO (General Purpose Input/Output) connector.

- GPS Neo 6M: it is a GPS (Global Positioning System) receiver with a ceramic antenna and a backup battery that can keep data. It has a 5Hz position update rate 50 channel receiver. The module interfaces with the raspberry via the I2C (Inter-Integrated Circuit) protocol, and it can be powered from 3 to 5 Volt. Using this GPS receiver it is possible to obtain the geographic coordinates of the place where the bicycle is located, in addition, other data such as altitude and time of detection are also available.

- MPU-6050: MPU-6050 sensor contains a 3-axis MEMS (Micro Electronic Mechanical Systems) accelerometer and a 3-axis MEMS gyroscope in a single integrated circuit. Thanks to the accelerometer it is possible to measure the acceleration in one direction. It uses a 16-bit analog to digital converter and can capture channels x, y, and z at the same time. The sensor interfaces with the raspberry via the I2C communication protocol. It can be powered with a voltage from 3V to 5V. Thanks to the three acceleration measurement axes, it is possible to calculate, with the appropriate physical and mathematical formulas, the degree of road slope as well as being able to calculate the lateral inclination of the bicycle. Furthermore, the sensor can detect ambient temperature.
- Reed LM393: LM393 and is equipped with a Reed contact (single contact normally open), a trimmer for adjusting the sensitivity, a status LED for the power supply, and an output signal LED. It allows obtaining a digital output signal depending on whether the Reed contact is open or closed. Just approach at a distance of at least 2 cm, a magnet to close the contact. The module interfaces with the raspberry via the I2C communication protocol and can be powered with a voltage from 3 to 5 V. Using two reed sensors, one mounted on the front wheel of the bicycle and one mounted on one of the pedals, it is possible to calculate various parameters useful for monitoring cycling activity: speed snapshot, distance traveled, and pedaling cadence.

Software Technology

- Mesh Network: it is a communication network composed of radio nodes organized in a mesh topology. In particular, it can be defined as an interconnection between MCU or MPU-based devices (nodes). If the nodes move constantly or frequently, the mesh increases its computational load. When a node stops working within the mesh, the other nodes continue to communicate, either directly or through other intermediate nodes. Wireless mesh networks can self-form and self-configure. A mesh network can be both WI-FI and Bluetooth. In the case in question, this network topology lends itself to the presence of temporary nodes in an area of the city;
- BATMAN (Better Approach To Mobile Ad-hoc Networking): is a routing protocol for multi-hop ad-hoc mesh mobile networks. On the BATMAN-based mesh network,

Originator Messages are sent at regular intervals, until they are received by the recipient at least once or until the packet is lost. Each packet loss allows you to go to find the best path from the node to node. Furthermore, the number of originator messages received by a given node is used to make a qualitative estimate of the route. The number of originator messages received is stored in a table (originator list) with which it is possible to calculate the best next-hop for the whole network. This feature makes the protocol optimized for use by cyclists circulating in urban areas.

10.2.5 Implementation

The implementation of the prototype focused on the construction of the device and its configuration. Each sensor is configured in order to acquire a different type of data. After the acquisition, the data are processed, shown on the device, sent to the mesh network, and eventually to the cloud. Once started, the device connects to the mesh network or a WiFi network by configuring itself. After the start-up phase, the user will see a screen like the one shown in Figure 10.8.

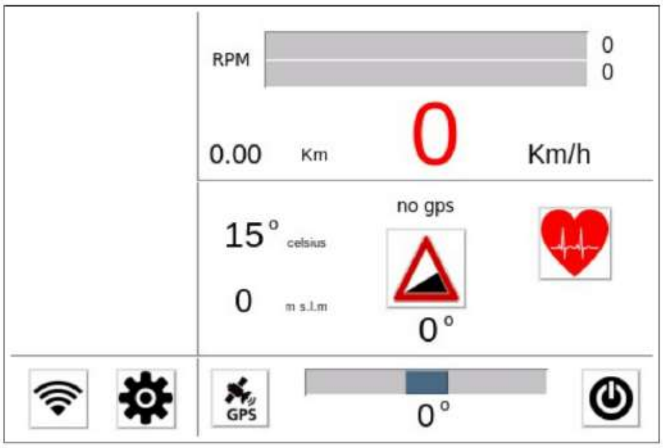


Figure 10.8: Main screen device.

From the main screen, the user can have various information and also enter personal data that the device will request to offer a better service. As the user proceeds with the march, the device shows information (Figure 10.9). Some of this information comes from direct acquisitions, while others come from the mesh network and/or the cloud. The mesh network is still capable of working without connecting to the cloud. The possibility of the nodes having processing capacity on the Edge allows obtaining information about the users who are using the device. This ability makes it possible to provide services that increase the personal safety of users by encouraging them to use the bicycle. The problem concerning this

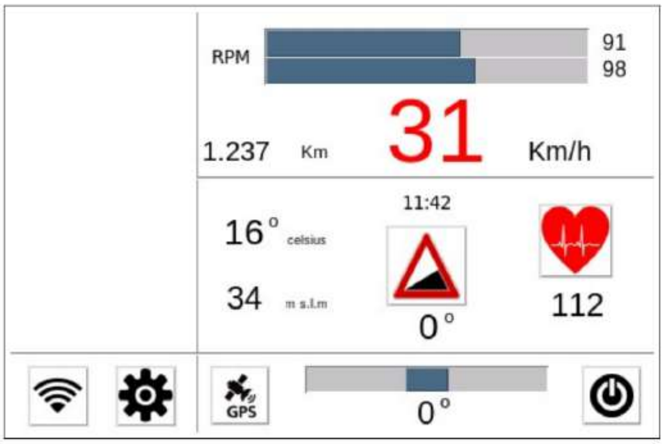


Figure 10.9: Screen device with data.

application, however, concerns precisely the intended use of network resources. To evaluate the impact of the use of resources for this type of application, benchmarks have been prepared for the evaluation of energy consumption. The assessments made will be discussed in the next section.

10.2.6 Test Results and Discussion

The experiments were carried out with the aim of demonstrating the possibility of using the Edge device in a real environment. Energy consumption and data exchange capacity were identified as critical aspects in the use of the device. Edge Device is a Raspberry Pi W zero powered by a battery with a nominal voltage of 3.7 V and a full load capacity of 3000 mAh.

Computational core consumption

The first test concerned the estimation of Raspberry consumption in different conditions of use and related connectivity activation.

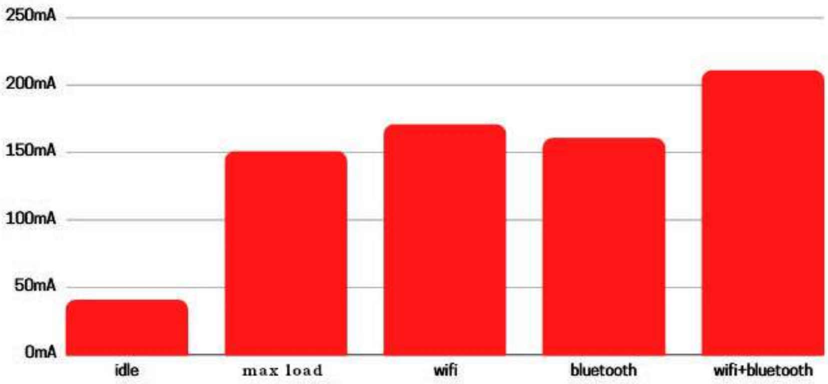


Figure 10.10: Raspberry consumption.

As shown in Figure 10.10 from test data carried out, it emerged that in idle mode the raspberry absorbs a current of 40 mAh, while with high CPU usage the device has an absorption of 150 mAh. If, in addition, the WIFI functionality is activated, the consumption is 170 mAh, while if the Bluetooth function is activated, the consumption is 160 mAh. If both functions are activated, the total consumption is 210 mAh. It was interesting to evaluate the consumption of the different sensors used in the prototype made. The results of the tests are shown in Figure 10.11.

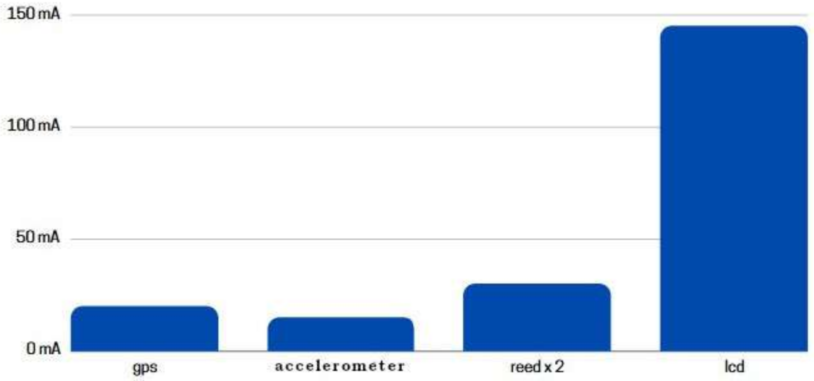


Figure 10.11: Sensors consumption.

The tests showed that the GPS receiver has a consumption of 20 mAh, the accelerometer of 15 mAh, while the sensor reed has a consumption of 15 mAh (considering that two are used, the total consumption is 30 mAh). The LCD touch screen has a power consumption of 145 mA. Based on the test results, in the worst-case scenario, in which the device works at full load with Bluetooth and WIFI connectivity and all sensors connected, the total current consumption will be:

$$210mAh + 20mAh + 15mAh + 30mAh + 145mAh = 420mAh$$

The battery has a full load capacity of 3000 mAh, this means that the device in the worst case can be used for a time of 7.15 hours. This usage time ensures its functionality for the use case under consideration.

Mesh network coverage

A tool called iPerf was used to evaluate the coverage of the WIFI mesh network, in particular its bandwidth and performance. The tests were carried out by varying the distance between the targets and evaluating the quality and the transmission rate of the signal even in the presence of obstacles. The tests in the absence of obstacles are carried out with devices placed each time at a distance of 1, 10, and 20 meters from each other.

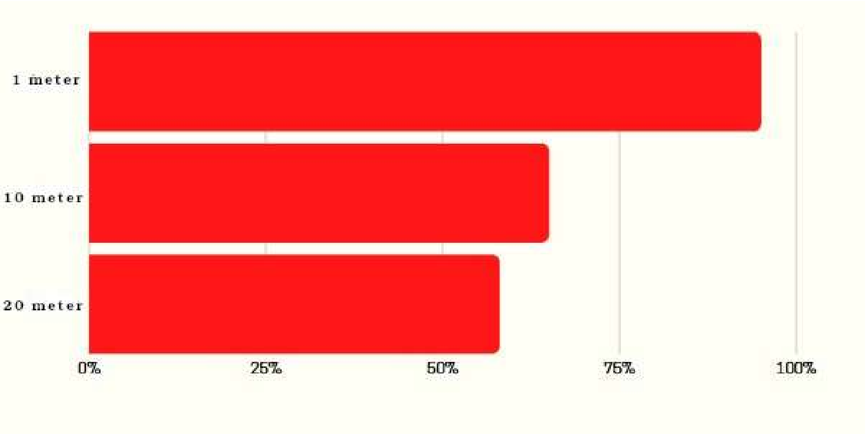


Figure 10.12: Signal quality without obstacles.

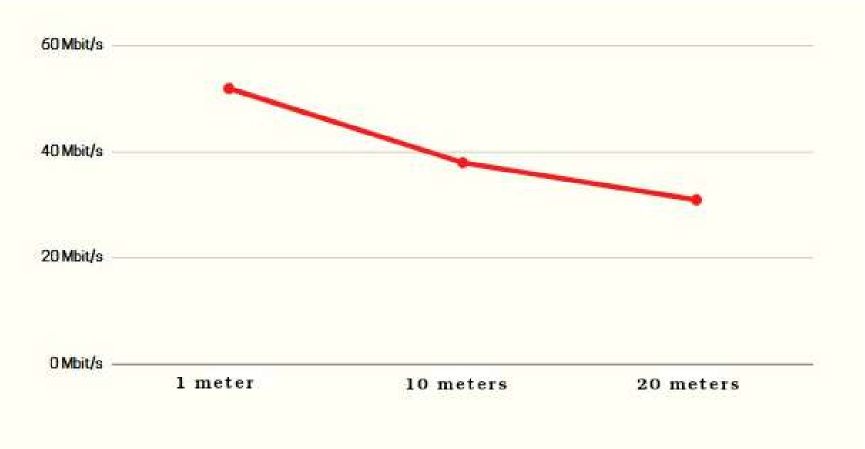


Figure 10.13: Transmission rate without obstacles.

As shown in Figure 10.12 and 10.13, if the devices are placed at a distance of 1 meter, the measured signal quality is equal to 95%, while the average data transfer speed is equal to 52 Mbit/s. If the devices are placed at a distance of 10 meters from each other, the signal quality is 65% and the average data transfer rate is 38 Mbit/s. With devices placed at a distance of 20 meters from each other, the signal quality decreases to 58% and the average data transfer rate drops to 31 Mbit/s. Another test phase involved the evaluation of the signal quality and the transmission rate in the presence of obstacles. The results are shown in Figure 10.14 and Figure 10.15.

With the devices placed at a distance of 2 meters from each other, and the presence of an obstacle of 25 cm, a signal quality of 80% and an average transfer speed of 47 Mbit/s were measured. In the case of devices placed at a distance of 6 meters, with the presence

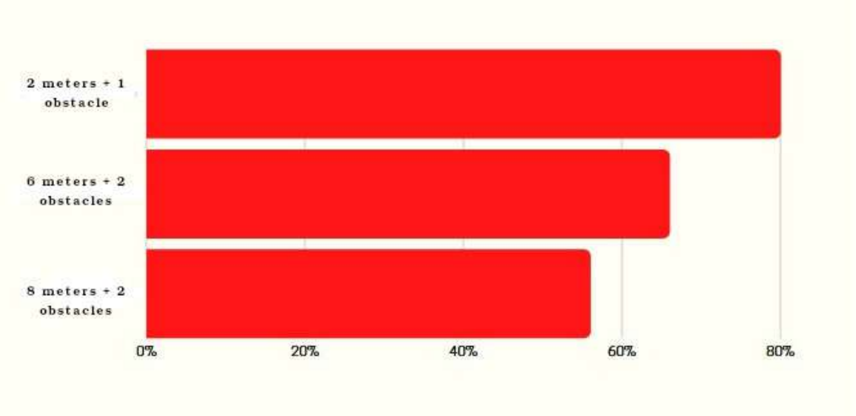


Figure 10.14: Signal quality with obstacles.

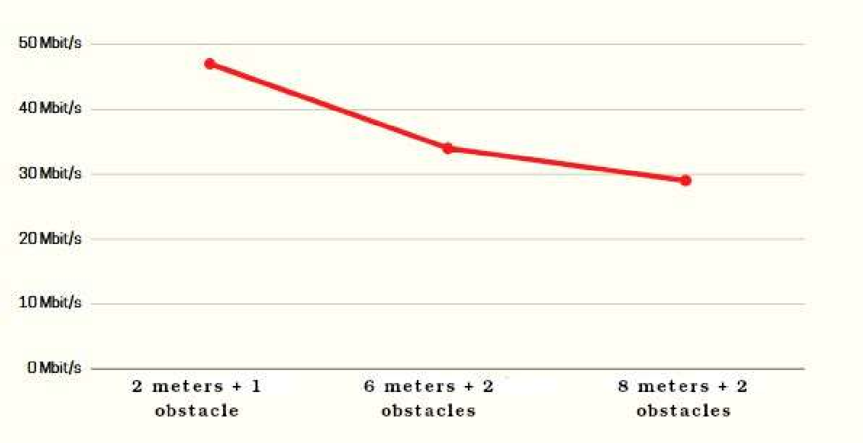


Figure 10.15: Transmission rate with obstacles.

of 2 obstacles each 25 cm thick placed at a distance of 1 meter, a signal quality of 66% and an average transfer speed of 34 Mbit/s was measured. While, when the devices are placed at a distance of 8 meters, with the presence of 2 obstacles with a thickness of 25 cm at a distance of 1 meter, a signal quality of 56% and an average transfer speed of 29 Mbit/s were measured. The results obtained allow us to conclude that the mesh network has good performance in open spaces without obstacles, which makes it ideal for being used as a network to interconnect the various bikes with each other.

10.2.7 Final Remarks

This section presents new technologies to increase the safety of bicycle users in urban areas. The key idea is to assemble a device with an electric bike to ensure continuous connection and collection of cycling performance data. The device is designed to create a mesh network with devices in use by other cyclists, thus creating ad hoc communications and interactions. We implemented a prototype of the proposed solution and performed some experiments, thus demonstrating its operation. Future works include testing the system by introducing data models based on the NGSI-LD protocol. The aim is to understand if and how complex data management can affect the performance of the device. Furthermore, it is interesting to replace the Edge device used in this paper with cheaper devices and compare the test results to identify a better solution in terms of system feasibility.

CHAPTER 11

Blockchain Technology for the reliability of the services

Blockchain technology represents one of the most innovative trends in various fields of application of modern computing. In fact, it opens up new scenarios in the field of the Internet of Transactions thanks to seven main characteristics: decentralization, security, consensus, immutability, transparency, responsibility, and programmability. One of its application domains is represented by the Smart Contract, an IT protocol aimed at digitally facilitating, verifying, and forcing the negotiation of an agreement between subjects without the need for third-party certification. Blockchain is one of the most promising technologies in the legal field. This chapter discusses various applications of the technology relating to the retail and transport sectors. The retail case analysis focuses on a system for tracking and validating the authenticity of tagged assets, based on a federated Blockchain environment. The federation is characterized by the presence of multiple audiences and private Blockchain instances. This solution allows one or multiple brands to store information within the private instance such as properties of the asset, history, maintenance, and more certificates. The public Blockchain is characterized by several brands sharing product information to prevent counterfeit goods from entering the market. Through federation, it is possible to cross-reference the data stored in different Blockchain instances to certify and validate the asset with additional levels of security. The analysis carried out in the case of "intelligent transport systems" aims to investigate the possible use and advantages of smart contracts to facilitate territorial continuity in transport to disadvantaged destinations. In particular, a case study of a hypothetical trip from Milan to the island of Lipari is analysed, also discussing a potential smart contract model.

11.1 Federated Blockchain Use Case

The Blockchain technology is increasingly used in the validation and tracking of assets [312, 313, 314] even to prevent counterfeiting phenomena [315, 316]. According to the Centerbrand institute for the fight against counterfeiting (INDICAM), [317] in the world there are about 461 billion Euros of fake goods, of which 85 billion Euros of fake goods are present in Europe.

The proposed work is intended to fight the counterfeiting of goods on world markets, providing a technological tool aimed at reducing the losses that this criminal system causes to companies. The counterfeiting market operates outside the legal business logic and estimating it is an arduous task. In this sector, there are no traces of financial statements, no taxes are paid and no data is shared on the generated profits. The counterfeiting affects all types of markets, in particular, that of high fashion and in general that of luxury goods.

The paper presents an innovative system for identifying and tracing assets, capable of storing information on the life cycle of an asset, including the process of sales, assistance, and possible resale of the product. Specifically, the proposed system is based on the use of the Blockchain, a technology for recording information recognized worldwide as inviolable and decentralized [318].

The system will allow luxury goods manufacturers to place their products on the market by certifying them as original. The certification will also be verified and validated in the resale of the product on the second-hand market, guaranteeing all users the guarantee of the product's originality. At the same time, it will be possible to trace the history of the product, identify its owners, buying and selling activities, assistance interventions, and so on.

The proposed method allows buyers who have purchased an asset to report the theft in the event of illicit appropriation of the asset. It will then be possible to start a product recovery procedure when its position is identified. The idea will make it possible to create a federated Blockchain system. We define it as *"a set of independent Blockchain systems managed by different companies that cooperate in order to share a part of their information that allows to obtain common advantages in terms of immutability and non-repudiation"*.

In our application scenario, a private Blockchain will be unique and internal to each company that owns the brand and will allow information on the product life cycle to be collected. The public Blockchain will be formed by different brands that will decide to join the system and will allow sharing of information on the product the detection of fakes placed on the market, or finding stolen products.

11.1.1 State of the Art

To the best our knowledge, at the time of writing, there are no specific initiatives related to the concept of federated Blockchain implemented for assets tracking. To highlight the differences with our research, a general overview of several Blockchain solutions in the assets tracking context is provided. In the last years, the Blockchain technology has been adopted in different fields, ranging from Industry 4.0 to Healthcare applications [319]. One of the main application of the decentralized system is the genuineness verification of an asset and its tracking along the supply chain. In [320], the assets are equipped with a particular packaging, and all the actions in the supply chain can be accounted for and logged with the timestamp. With the proposed solution, the path of the products in the supply chain is tracked, guaranteeing their authenticity, and identifying the exact point of failure in case of tampering.

The potential of Blockchain technology is also applied in the Art Industry. Specifically, ArtChain [321] is presented as an integrated trading system, in which the origin and the traceability of artistic resources are recorded. In particular, with the use of the Proof-of-Authority (PoA) model, museums and art galleries act as internal authorities within the ArtChain network. A Blockchain-based luxury goods certification service was developed by Luxochain [322]. It offers a digital passport to uniquely identify a luxury item, certifying its originality. Each product has a "Digital Twin" registered for life on the Blockchain, and therefore not editable. Luxochain provides a smartphone application through which each customer can know if the product is authentic or counterfeit, and know its history.

Furthermore, Blockchain technology is also used in the wood sector to ensure the transparency of data management in a supply chain. An asset tracking solution to fight illegal logging in Peru by preventing users from breaching data and information on certified trees is discussed in [323]. Specifically, this application is based on the storage of data, for example, photos of trees in the different phases, certificates, and geolocalization within records that are stored as transactions in a distributed ledger. The problem of tracking the origin of a product and maintaining ownership of digital assets using the Ethereum smart contract is discussed in [324]. In the implemented network, the parties track the source of the products managed by the nodes increasing the visibility of the system. The set up a secret Blockchain system to solve the problems of digital content exchange is discussed in [325]. Each digital content is encrypted and provided with a fingerprint so that in the event of illegal loss of the product, the destination is traced. The On-Chain network, using a license, can access the digital content

and carry out a verification process through consensus algorithms.
The tracking of products and resources in a supply chain is also done through the innovative development of physically non-clonable functions (PUFs) combined with the [326] Blockchain. PUFs are devices that provide a unique fingerprint to a product. The weakness of this technology is that it can only be applied to electronic devices or assets with power supplies. A patented [327] asset tracking verification system allows the authentication of a product by identifying the physical components of the asset using an image. Once the system has acquired an image via the smartphone, it analyzes that to identify the physical characteristics and the unique identifier of the target asset. Differently from the above mentioned recent scientific initiatives, this section proposes an innovative system of identification and tracking of assets through a federated Blockchain system. The private Blockchain will allow the collections of information on the product life cycle while, the public Blockchain will be formed by different brands that will decide to join the system, and will allow you to share information on the product that can allow you to detect fakes placed on the market, or find stolen products.

11.1.2 Motivation

Blockchain technology has been recognized as inviolable and trusted [328]. Through distributed ledgers, a new world economy based on cryptocurrencies was defined, and a new concept of tracking immutable and decentralized data was born. [329]. Nowadays, one of the major problems faced all around the world is the counterfeiting of goods, e.g., luxury goods, food, or technology. As described above, to contrast this problem, it is possible to use the Blockchain technology for tracking and validating the authenticity of goods. The current paradigm of the world market is multi-brand, in which different companies from different countries cooperate together to create products of highest level [330]. Through conventional Blockchain platforms, it is possible to guarantee the certification and originality of an asset and to collect information on the life cycle of the product. However, one of the major weaknesses of such an approach is the lack of interaction between multiple Blockchain systems that allow guaranteeing an additional level of certification. Here is where the Blockchain federation idea comes from. Specifically, the basic idea is to create a single interoperable federated Blockchain certification ecosystem that is distributed across multiple independent brands for the management of products/goods. The federated environment is structured in such a way that each brand has its own private and public Blockchain: i) with the private Blockchain the active interactions between the brand and the distributors are managed, and finally, the products are certified and validated; ii) the interactions (purchase/sale/maintenance) of an asset with the end customer and the Product life-cycle are managed with the public Blockchain. Blockchain federation enables the interaction between the different systems of multiple brands for sharing information. This sharing is carried out by a shared protocol that allows each private Blockchain system to work autonomously, but at the same time to share useful information with a public federated Blockchain ecosystem. The latter needs only part of the information of the private Blockchain system to guarantee the immutability and non-reputation of the whole good's life-cycle.

11.1.3 System Design

With this work, we propose a solution based on a federated Blockchain environment to create a system for tracking and verifying the authenticity and ownership of assets and luxury goods. The Blockchain Ecosystem is structured as follows:

- **Private Blockchain:** it is unique and internal to every single company owner of the brand and allows the collection of information on the product life cycle;
- **Public Blockchain:** is characterized by different brands adhering to the system, and it allows to share information on the product in order to detect fake assets placed on the market and to find stolen goods.

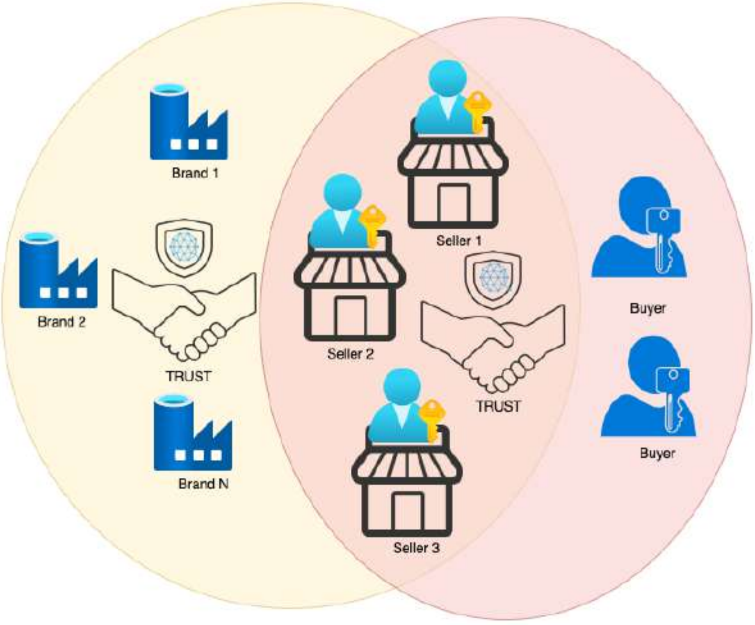


Figure 11.1: Reference scenario.

Figure 11.1 shows the reference scenario of the asset tracking model based on the federated Blockchain. *Brands* are the companies that produce the assets. Each *Brand* keeps track of the entire history and life cycle of the product within its Private Blockchains. From a technical point of view, the idea is to insert an NFC tag in the asset. The same can be replaced with a QR_Code to allow the compatibility with devices that are not equipped with NFC technology. The NFC tag or QR_Code is associated with a Universally Unique IDentifier (UUID) which is registered in the Brand's Private Blockchain. The manufacturing company (*Brand*) is the first owner of the product. In the case of purchase, the *Seller* is registered within the Public Blockchain and becomes its official owner, and the purchase is legitimized by a smart contract, which is generated between the retailer and the *Buyer* through a mobile application, using an authentication system multifactor. Every time the ownership of good changes, the Smart Contract is responsible to redefine the ownership and authenticity.

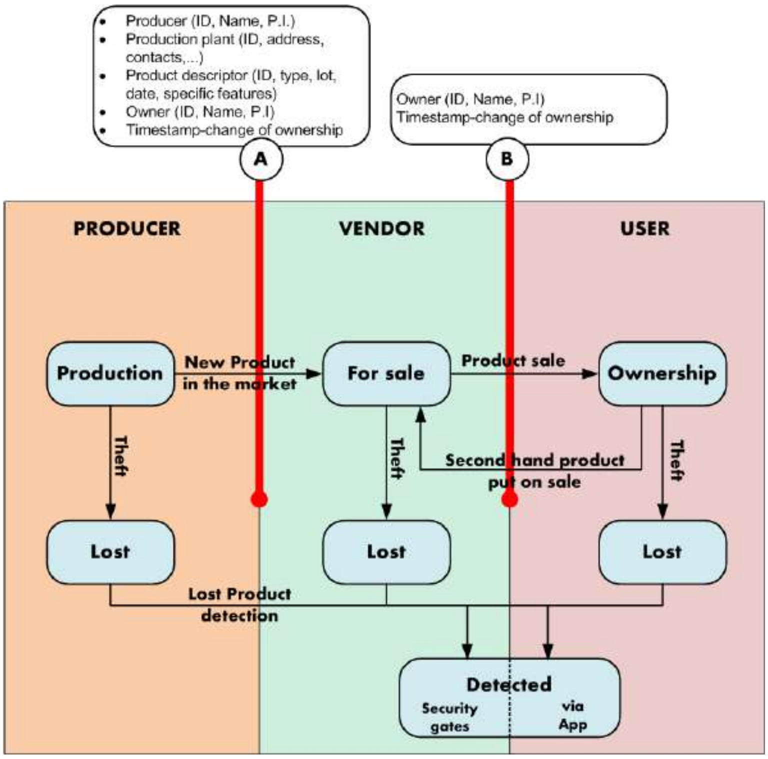

Producer (ID, Name, P.I.)
Production plant (ID, address, contacts,...)
Product descriptor (ID, type, lot, date, specific features)
Owner (ID, Name, P.I)
Timestamp-change of ownership
Owner (ID, Name, P.I)
Timestamp-change of ownership
A
B
PRODUCER
VENDOR
USER
Production
New Product in the market
For sale
Product sale
Ownership
Theft
Theft
Second hand product put on sale
Theft
Lost
Lost
Lost
Lost Product detection
Detected
Security gates
via App

Figure 11.2: Product Life Cycle diagram.

Figure 11.2 reports the life cycle of the product by identifying the actors involved in the proposed system (i.e., manufacturer brand, vendor, and end-user), the states in which the product can be found (i.e., production, for sale, ownership, ...) and the possible activities carried out on the product that will be registered in the system. In particular, the diagram highlights the transitions A and B which correspond to the invocation of a Smart Contract

respectively between Brand and Vendor, and between Vendor and User.
The owners of the goods, through a mobile application, can scan the NFC tag or QR_code and verify the authenticity, history, warranty, and ownership of the product. In the event of theft, following a complaint by the owner, the artifact is registered in the Blockchain as a stolen asset. When the code of a stolen asset is scanned, a notification is triggered to alert the person in possession of the asset, and the good can be traced and recovered.
The proposed system allows us to guarantee and certify the resell of a used product. In this case, a Smart Contract is signed between the Buyer and the Seller. The history and the transfer of ownership of the asset itself are traced, providing a used warranty.Through a mobile application, which will be able to read the NFC tag or QR_code, it is possible to ensure that:

- The object is original as guaranteed by the Blockchain;
- The identity of the owner is certain, and any maintenance under warranty is known;
- The warranty status of the item in order to avoid fraud by customers;
- In case of theft, whoever initiates the authenticity verification procedure allows tracing the good.

The user can resell his object to official retails, which will later be offered for sale as used, but with the guarantee that it is an original not counterfeit good. The change of ownership within the Blockchain is a guarantee of the absence of scams. In addition, by providing the free app service in the stores, the user can verify the product, warranty, and authenticity directly with his smartphone.
In the case of a fake article, the app will detect the object that does not belong to the Public Blockchain and therefore to that company. Otherwise, the user will have information on the life of the product, whether it is new or used. Possible advantages deriving from the proposed system are:

1. the user will not have to keep the receipt for the warranty, the Blockchain guarantees the purchase date;
2. the company that produces the product will be able to reduce the losses on the balance sheet due to counterfeiting;
3. the company that produces the product will be able to open and/or expand the second-hand market on its goods, having positive returns in terms of respect for the environment and reduction of unbridled consumerism;

4. with the improvement of the lifestyle of the population, expensive items can be purchased as used in the original version and therefore those who live in unfavorable conditions will have access to good quality products at a lower price.

Use Case Scenario

A customer who wants to buy a Louis Vuitton brand bag goes to a retailer and chooses the desired product. This object is equipped with an NFC device and is registered within the Blockchain as property of the brand. During the purchase of the product, the customer signs a smart contract with the retailer through the app. The customer's ID is registered inside the smart contract during this phase, while in the case of a gift to third parties, the third person's ID is registered. The product and its ownership are registered within the private Blockchain and, only after this procedure, the buyer becomes the official owner of the asset. When the product is sold or transferred to third parties, the owner will have to go to the nearest retailer, a member of the public Blockchain, to stipulate a new Smart Contract and register a new transaction on the Blockchain.
Furthermore, the system will be equipped with an app through which the owner can verify the authenticity and history of the product. If the product has been stolen and resold, the owner will have to report the theft on the system, and that product will be "reported" within the public Blockchain.
When a customer tries to verify the authenticity of the product and it is marked as stolen, a process of tracing and recovering the product will be activated as it will be possible to trace the person who holds the stolen product. Each retailer belonging to the public Blockchain will be equipped with an NFC detection system at its entrance, which will scan each product, and if it is recognized as stolen, an immediate product recovery procedure will be started. This system will limit the counterfeit goods market, as it will be cheaper to buy an original used product rather than a very low-quality product without certification.

11.1.4 Implementation

In this section, we discuss a prototype implementation of the federated Blockchain Environment. We choose to use Ethereum [331] as it is the most used platform and allows the development of Distributed Apps (DApps) using Smart Contracts [332].

Trusted Federated Environment

Three strategies were used to implement the system prototype:

1. **Full Ethereum Public Network Node**: each Brand stores a copy of the entire Blockchain data and participates in the mining process for the consensus mechanism;
2. **Etherum Private Network**: each Brand implements a local node with the benefit of Ethereum frameworks such as Truffle and Ganache, reducing the disk space required, the latency, and global mining overhead of the Public Network implementation;
3. **Web Server API**: each Brand implements a web-based Application Program Interface (API) based on Infura to interact with Ethereum to reduce latency times for recording information within the distributed registers [333].

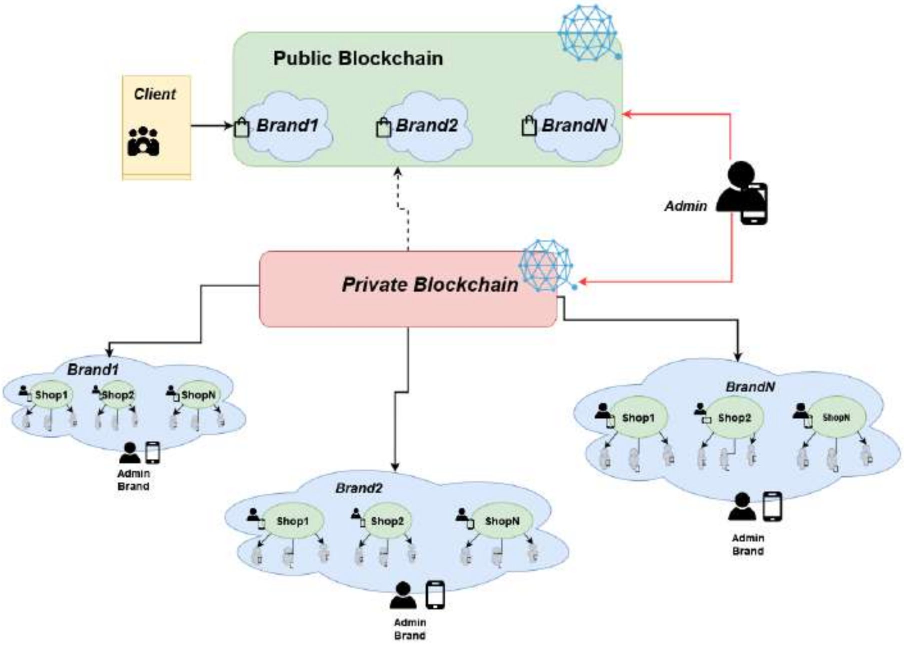


Figure 11.3: System Prototype.

Figure 11.3 shows a Block Diagram representative of the system at a macroscopic level:

- **The Public Blockchain** is implemented through an Ethereum network, where each node is characterized by a Brand;
- **The Private Blockchain** is a private instance of Ethereum internal to every single Brand. The nodes of this distributed ledger are the individual stores and the retailers affiliated with the system;

- **The Brand Administrator (Admin)** is the entity that has access to both the private and public Blockchain;
- **The customer (Client)** is the entity that becomes part of the federation when a product is purchased and a smart contract is signed between the two parties.

The implemented system is RBAC (Role-based access control) [334].
Specifically, we use the FIWARE IAM [335], with Keyrock and Wilma modules, for the management of registered users within the organization. The user can register by specifying his status: brand manager, supplier, store owner, employee, and his identity will be validated by users with special administration roles.
A general user *consumer*, will not have to register within the system. User registration is required for access to the management area where it is possible to view all products, register sales, or maintenance. Each product registered in the private Blockchain will be equipped with a QR_Code or an NFC tag. A dedicated app allows the users to frame the tag and obtain information about the product.
In case of purchase, the buyer can purchase the product and subsequently sign the smart contract. With the signing of the smart contract, an exchange of ownership of the asset is defined within the public Blockchain [336]. The owner of the product becomes the buyer and no longer the brand.

Figure 11.4 shows a Block Diagram representative of the system at a microscopic level.
The system starts from the different private Blockchains of the individual brands. One or more brand admin users, *admin brand*, authorize the shop owner to enter both the public and private Blockchain. From this moment on, all the smart contracts generated between the various actors (admin or client) for the single shop will be transcribed in whole or in part on the public Blockchain. The shop owner, or in any case an admin shop, can appoint a special user for his shop. The special user is a shop assistant, or an employee enabled to generate smart contracts with customers in place of the admin shop.
Customers (users) who buy a product become owners of it. A real-time generation procedure of the smart contract is activated through the app (Figure 11.5). The protagonists of this procedure are the customer who has downloaded the app on their smartphone, the special user, or the shop administrator.
In addition, the operators of the shop (admin shop, special user, shop assistant) through an interface (API) with their management systems (product management) will also be able to manage any guarantees or changes on the products by writing everything on the Blockchain.

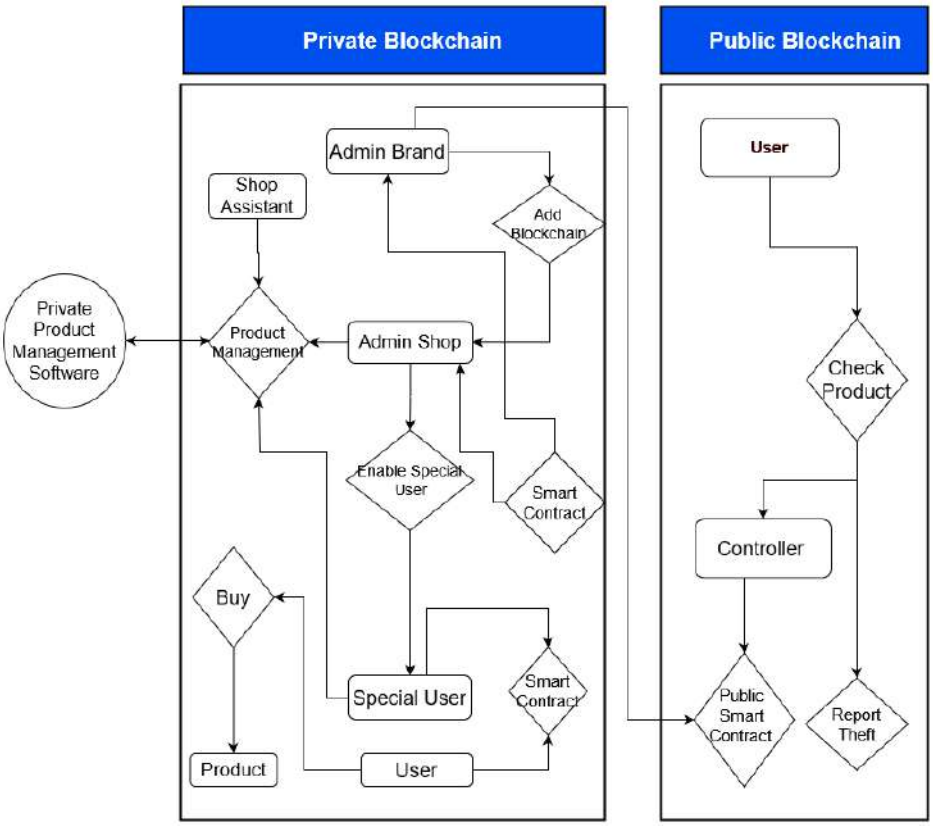


Figure 11.4: Macroscopic System Representation.

As for the public Blockchain, any user (user) can check the info on a specific product, or report that the product owned by him has been stolen.

Smart Contracts

Figure 11.5 depicts a Flow Chart containing the procedure for changing ownership of objects with the signature of the smart contract. When a user requests to sign a smart contract, the interested party will be "notified".

If the interested party gives the authorization to close the smart contract, the latter is signed and written on the Blockchains, otherwise, the procedure is closed. Once the signature request has been made, it must be closed in a limited time, after this time the procedure is closed.

11.1.5 Experiments

In this section, we analyze the response time of the deployed system, to validate the usability of the federated Blockchain approach in real-world applications.

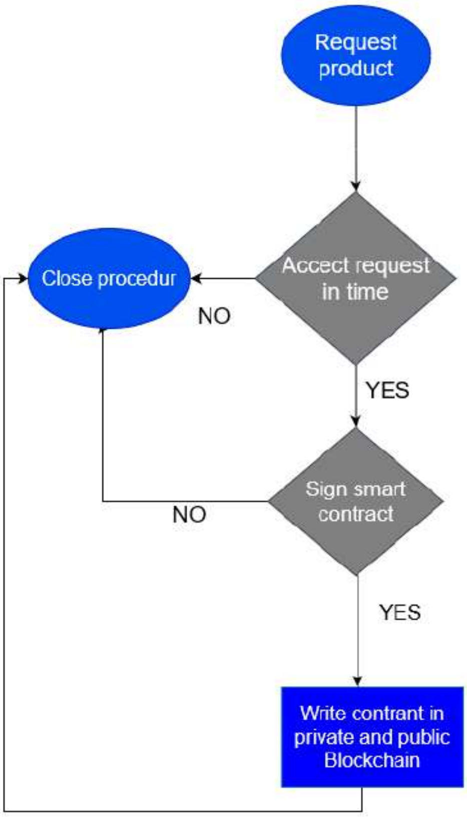


Figure 11.5: Smart Contract - Process of changing ownership of an asset.

Testbed Setup

The proposed system is implemented using two Virtual Machines (VM) hosted on the Garr network and a Raspberry Pi4. The back-end of the system is deployed on a VM with the following system specification: Intel Xeon E3-12xx v2 (Hivy Bridge, IBRS), 2 GHz, 4 GB RAM running Ubuntu Server 18.04, and 4 CPU cores. Furthermore, we used another machine with the same characteristics for the implementation of the Public and Private Blockchain instances.

For the emulation of the asset's certification process, we used an Edge device emulated with a Raspberry Pi4 Broadcom BCM2711, Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz. Each test has been repeated 30 times considering 95% confidence intervals and the average results are plotted.

Performance Analysis

The performance analysis aims at verifying the data exchange system for the communication with the Blockchain based on a web server approach comparing Public and Private Ethereum networks. RPIs 4 represent the device of the actors of the system. A summary of experiments setups are reported in Table 11.1.

Table 11.1: Summary of experiments performed.

Parameter	Value
Test executed for each scenario	30
Confidence interval	95%
Gas used by transaction	833,105
Gas price (Gwei)	200
Average cost per transaction (ETH)	0.000166621

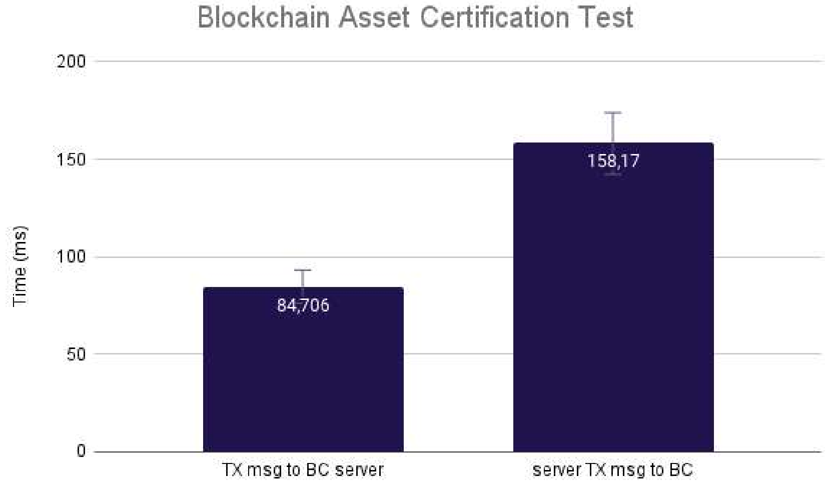


Figure 11.6: Blockchain Data Writing with Web-Server Approach.

Figure 11.6 depicts the average execution times of the algorithms implemented for the certification of an asset on the Blockchain. The average time for transmitting data relating to transitions, verification, purchases, or sales to the server is 0.084 seconds.
To write data from the Web Server to the Blockchain, the implemented algorithm requires an average execution time of 0.158 seconds. From the analysis carried out, it is clear that going to send encrypted data to the web server allows us to have faster system performance than certifying the data directly on the Blockchain.

In the Ethereum Public Network, products are registered in the public network, for each brand, subject to high costs and transition times. In the private network, each product is registered in an instance consisting of a single node and a single hash code, calculated as the MD5 of the transaction hash.
Using this approach, it is possible to obtain shorter waiting times and a negligible ETH cost, which results in an immediate cost saving for the overall system implemented.

Figure 11.7 shows the difference between the two approaches in terms of extraction time expressed in seconds.
On the x-axis, we report the number of simultaneous product registration requests, from 1 to

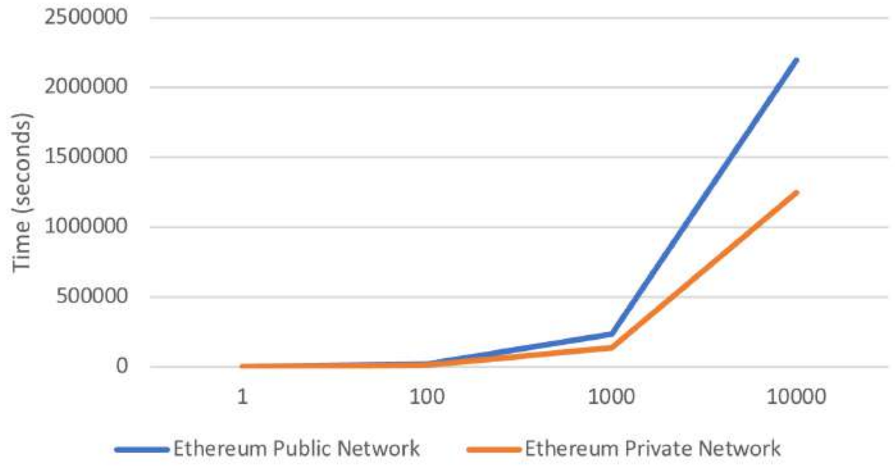


Figure 11.7: Time comparison for Public and Private Ethereum Network approaches for a variable number of product registration request.

10,000. The processing time expressed in seconds is shown on the y-axis. The graph shows that for a small number of simultaneous requests, both implementations present similar behavior.

After almost 100 requests it is possible to observe a significant performance degradation for the Ethereum Public Network implementation. This is caused by the mining queue waiting time that is obliged despite the high Gas price (200 Gwei) adopted to allow a faster processing. The performance for Private Ethereum Network maintains a linear time increase, as expected because the mining queue is not susceptible to global network congestion.

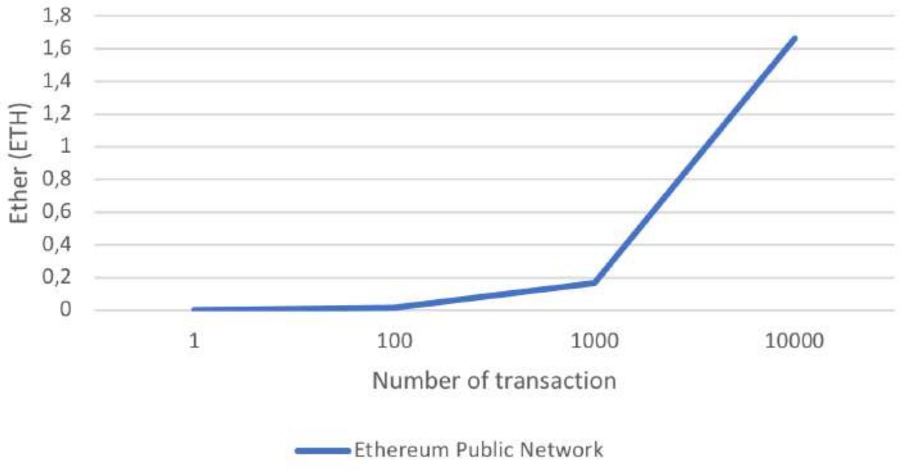


Figure 11.8: Cost analysis.

Figure 11.8 highlights an economic analysis in terms of Ether crypto coin (ETH) to demonstrate the investment required to maintain a fast and secure infrastructure.

Considering the average cost of a single transition allowing to register the product in the Public Ethereum Blockchain, this approach is doable for an expected low number of products.
The Private Blockchain approach is recommended for a high number of transactions and therefore appears to be the best solution for our system as it allows exponential cost savings.

11.1.6 Final Remarks

In this section, we proposed an innovative method for tracking assets through a federated Blockchain system. The system was designed considering the use of Public and Private Blockchain instances to identify the best solution in terms of scalability. In addition, a prototype was implemented for the registration of an asset with a tag, through smart contracts. The system was tested in order to evaluate its usability within the reference scenario. In future work, we propose to have more private and public Blockchain instances communicated, in order to test the patented stolen asset tracking features.

11.2 Smart Contract in Urban Mobility Scenario

Blockchain technology represents one of the most innovative trends in various fields of application of modern computing. In fact, it opens up new scenarios in the field of the so-called "Internet of Transactions" thanks to seven main characteristics: decentralization, security, consensus, immutability, transparency, responsibility and programmability. Specifically, the Blockchain is a data structure characterized by an incremental list of records, called "blocks", securely interconnected with each other by means of particular encryption functions. Each block has a cryptographic hash value of the previous one, a "timestamp" and transaction data. Since each block contains information about the previous one, they form a chain in which each new block is connected to the most recent one. Therefore, blockchain transactions are irrevocable as, once recorded, the data of a particular block cannot be retroactively altered without modifying subsequent ones. Blockchain technology was initially created for the management of cryptocurrencies (such as, for example, Bitcoin) and cryptographic "tokens". Blockchain has gradually become the basis for a new way of thinking, organizing and managing relationships between different subjects, introducing a new concept of global trust. In order to explain the trust problem, let's consider a man who often uses his credit card to pay for goods and services. The shopkeeper allows the purchase of the goods only after having ascertained that the buyer has the sum to be paid. To ensure this, the credit card is inserted into the card reader and a verification process is carried out through the bank. Once completed, the bank will finally make the payment and issue the invoice. From this example it can be seen that thanks to banks acting as intermediaries, there is trust, which makes it easier to purchase goods and services. In fact, a simple way to solve the trust problem is to use an intermediary. Taking the previous example into consideration, the bank performs this function by facilitating the transaction. There are also other ways to solve this problem by using a distributed ledger and combining it with consensus algorithms. So, instead of managing the transaction on a single broker, multiple parts of the network come into play, who are aware of all the transactions stored in the distributed ledger. Therefore, each party knows exactly how much the customer can spend. Blockchain technology follows this distribution and consensus method, and is a network that uses a decentralized and publicly verifiable ledger to solve the problem of trust. Blockchain leverages and combines the capabilities of a computer network with cryptographic technology to store and process data. Any computer in a Blockchain network is known as a node and can be located anywhere in the world. A Blockchain, in general, register, processes, saves and verifies transactions based on the concept of shared accounting, in which each node in the network stores the same copy of the register, and therefore all the transactions that take place on the platform. This decentralized (shared) aspect of the ledger makes the Blockchain a public, complete, permanent, and verifiable authority for administering and storing transactions. Thanks to its characteristics, Blockchain is, nowadays, a technology capable of addressing problems of access and management of information in different application domains, including, for example, financial services, personal identity security, healthcare , logistics, public administration, communication, the definition of ownership deeds and certificates of authenticity (from the English “non-fungible token”) and the intelligent contract (from the English “smart contract”). The latter consists of an IT protocol aimed at facilitating, verifying and digitally forcing the negotiation of an agreement between subjects without the need for third-party certification. For this reason, Blockchain is one of the most promising technologies that could revolutionize the legal field, paving the way towards the concept of "legal engineering". We can define legal engineering as "a practice that allows theoretical legal aspects to be implemented on an IT system to support legislation, contributing to the drafting of new laws, legal rules, and/or legal documents, facilitating the interpretation of existing ones and avoiding ambiguity, so that they adapt to the progress of Information and Communication Technology (ICT)”. This section is configured in the context of "intelligent transportation systems" (from the English "Intelligent Transportation systems (ITS)"). This section is configured in the context of "intelligent transportation systems" (from the English "Intelligent Transportation systems (ITS)"). The objective is to investigate the possible use and advantages of the smart contract to dynamically establish agreements between the traveler and various transport companies in order to facilitate territorial continuity with disadvantaged destinations such as the Mediterranean islands. In particular, the case study of a hypothetical journey from the city of Milan to the island of Lipari (one of the Aeolian islands in the municipality of Messina) will be examined.

11.2.1 State of the Art

Smart contracts can find a wide spectrum of potential application scenarios in the digital economy and smart industries, including financial, management, healthcare and Internet of Things services, among others, and have also been integrated into major platforms development systems based on Blockchain, such as Ethereum and Hyperledger [337]. For example, they have been applied to allow interaction with the electronic voting system for the management of political elections [338]. This was possible through the specification of an IT architecture designed specifically for electoral processes, evaluating Blockchain technology as an alternative to current voting systems. Effectively managing the healthcare procurement process is critical for healthcare providers. Despite significant advances in new information technologies, the procurement and distribution processes of medical supplies are still not optimal and their contracting generally takes a long time. These limitations have been overcome by the use of smart contracts as discussed in [339]. Smart contracts have been used to ensure reliable sharing of electronic health records between different entities involved in a smart healthcare system [340, 341]. Specifically, four forms of smart contracts have been proposed for user verification, access authorization, misbehavior detection, and access revocation, respectively. The use of smart contracts in a smart agriculture scenario with the aim of providing reliable data to farmers and other related users on a single integrated platform is discussed in [342]. Projects financed by government funds typically require the use of cumbersome tender procedures that can sometimes lead to corruption due to lack of transparency. In this context, smart contracts have been proposed in order to ensure simple and transparent administration of public projects that allows various stakeholders to control the entire review and financing process [343]. The land registry system is a time-consuming process that typically involves many intermediaries, thus increasing possible fraudulent cases. These problems can be eliminated by using Blockchain technology and smart contracts for land registry management as discussed in [344]. The use of smart contracts also finds application in the field of renewable energy as discussed in [345]. Specifically, a Peer-to-Peer energy trading system based on smart contracts is proposed, which offers consumers the opportunity to contribute to the grid through renewable energy sources. Smart contracts are also used in the field of intelligent transportation. An electronic payment system for charging electric vehicles using smart contracts is discussed in [346]. Such technology is used to control and manage payments and to decentralize the payment system so that devices can automatically pay each other. Furthermore, the system also allows sharing of private charging facilities by automating payments. A model of an ecosystem for on-demand mobility, which enables easy, fast and reliable transactions by leveraging artificial intelligence and smart contracts is discussed in [347]. With the rapid development of the transportation industry, ITS is trying to quickly adapt to industry changes through building new organizational forms, management methods and incentive mechanisms. However, many new administrative problems have arisen due to the complexity, diversity and uncertainty of the transportation system as discussed in [348]. A decentralized autonomous organization that leverages smart contracts can potentially solve problems arising from a complex transportation system, but at present there is no system capable of providing this type of service to travelers in Italy and Europe. At present, from our analysis of the literature, there is still no application of smart contracts in the transport sector to facilitate territorial continuity and this article intends to fill this gap.

11.2.2 Smart Contracts to Facilitate Territorial Continuity

The case study examined is a hypothetical journey starting from the city of Milan and destination the island of Lipari, one of the Aeolian islands of the municipality of Messina in Sicily (Figure 11.9). In Lipari there is no airport, and the only maritime connections are

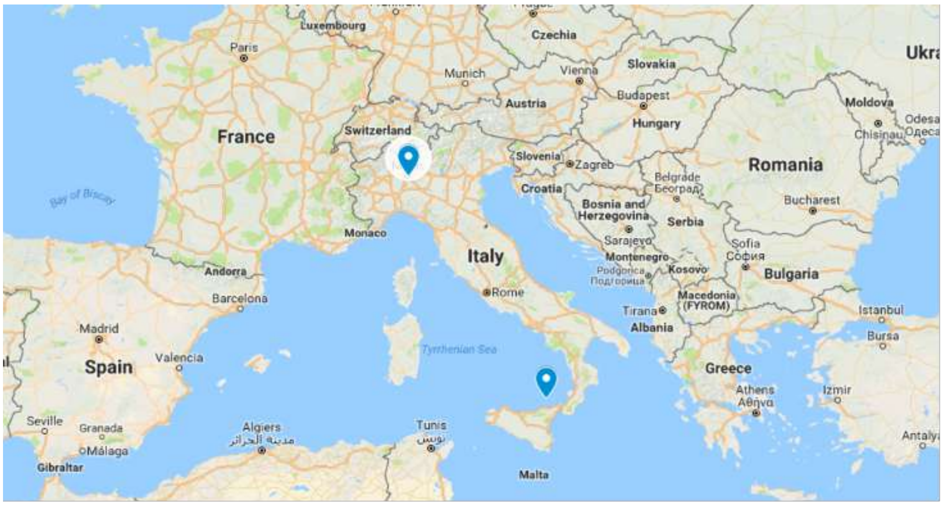


Figure 11.9: Departure and Destination Locations

through ferries or fast ships that connect the island to Sicily, Calabria or Campania. It is therefore easy to understand why there cannot be a single transport ticket that allows you to connect the island with a location in northern Italy. Therefore, different means of transport (with different travel tickets) will be necessary to cover the different routes necessary to reach the destination. In the example below, starting from Milan, it will be necessary to purchase several tickets:

1. urban ticket to connect the city center with Linate airport - ATM company;
2. flight ticket Milan-Catania - ITA Airways company;
3. interurban ticket from Catania-Milazzo airport (ME) - SAIS Autolinee company;
4. Milazzo-Lipari hydrofoil ticket - Liberty Lines company.

As can be seen, the minimum number of public transport vehicles needed to reach Lipari starting from Milan is four, and all tickets are issued by as many companies which often do not have any commercial partnership agreement. At present, organizing a trip presents four main critical issues:

- if you do not know the area, it is necessary to invest considerable time in research to identify all the companies that cover a portion of the section affected by the trip;
- there is no single transport contract between the traveler and the various companies involved;
- if a traveler misses a connection due to a cancellation or delay on the previous route, he will have no protection and will lose the tickets for the subsequent routes, with the need to reschedule and repurchase the remaining part of the trip, suffering clear economic damage;
- consequently there is no guarantee on the total route and reaching the final destination.

This is a typical example in which it would be possible to exploit Blockchain technology for the creation of a smart contract in order to guarantee an agreement between the final traveler and the numerous companies involved, offering a simple, reliable, flexible and secure service. Obviously, it would be necessary for the various transport companies involved to join a Blockchain system with private authorization for the dynamic "smart" creation of travel tickets. Thanks to smart contracts, the traveler and the various companies involved could dynamically establish, based on the desired itinerary on a given date, a smart contract at the time of need. This approach could potentially overcome the limits on the timing of stipulation of traditional contracts which typically occur a priori, with obvious application limits in scenarios that require a high degree of dynamism. Furthermore, both the traveler and the transport companies involved would have the certainty that a particular carrier could not unilaterally change the travel conditions without the other actors involved becoming aware of it. For example, one of the possible travel conditions that could be applied in a contract between the parties could be: the release of a single cumulative ticket, a timely "re-routing" on an alternative means of transport in the event of cancellation or delay on a route. Can be added the reimbursement of food and accommodation costs if the alternative means of transport were to leave the following day, with consequent rescheduling of the tickets for the remaining routes. In alternative can be evaluated the reimbursement of the entire cumulative ticket, if the traveler were not able to reach the destination within the established times . This approach would, in fact, guarantee a high level of quality of service (QoS) which can lead to preferring a solution that offers greater guarantees compared to competing companies that are not able to offer the same travel conditions and guarantees. The general objectives of using smart contracts to support transport in a context of territorial continuity are:

- satisfy common contractual conditions (such as, for example, payment terms);
- reduce both malicious and accidental complaints to a minimum;
- minimize the need for trusted intermediaries.

Related economic objectives include reducing:

- damages from fraud;
- arbitrations;
- judicial costs and other transaction costs.

11.2.3 Smart Contract Model

From a technical point of view, a smart contract of this type would be nothing more than the "translation" or "transposition" of a contract into computer code which, once executed, would allow:

- automatically verify the fulfillment of certain conditions (checking the basic data of the contract);

- automatically self-execute actions when the conditions determined between the parties are reached and verified.

The software is based on a code that "reads" both the clauses that have been agreed. The operating conditions in which the agreed conditions must occur and is automatically self-executing when the data referring to the real situations corresponds to the data referring to the conditions and the agreed clauses. In this case, the various clauses would be implemented as "blocks" of the Blockchain which would guarantee trust, reliability and security which in traditional contracts must necessarily be delegated to a "third" part (for example a notary). Consequently this flow allowing transactions to be carried out without intermediaries . The Smart Contract model described could be extended to various air, naval, rail and road transport companies (buses and taxis) as shown in Figure 11.10. Based on this model, the user or tour operator searches for the various available combinations on a travel web portal for booking travel tickets. Once the preferred combination has been chosen, the web portal, interacting with a public Blockchain system, will dynamically request with an "on demand" approach the stipulation of the required Smart Contract. In the smart contract involving both the traveler and the chosen carriers. Once this is done, the users will proceed with the payment of the cumulative ticket and with the traveler’s registration in the communication systems of the chosen travel companies. This is necessary in order to provide they with updates on the trip in real time via text message or dedicated mobile app.

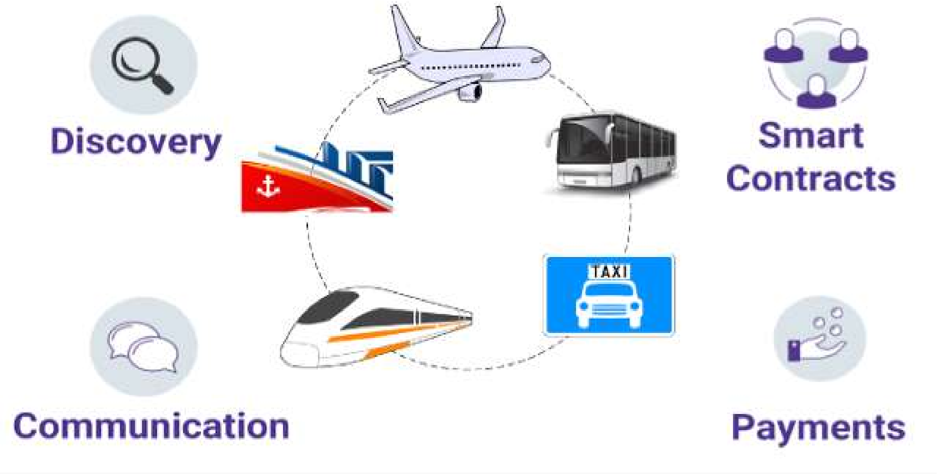


Figure 11.10: Smart Contract model applied to transport to guarantee territorial continuity

11.2.4 Final Remarks

Blockchain technology represents one of the most innovative trends in various fields of application of modern computing. One of its application domains is represented by the Smart Contract, an IT protocol aimed at digitally facilitating, verifying and forcing the negotiation of an agreement between subjects without the need for human third-party certification. Therefore, Blockchain is one of the most promising technologies in the legal field. In this section we wanted to focus attention on intelligent transport systems, with the aim of investigating the possible use and advantages of Smart Contracts to facilitate territorial continuity in transport with the Mediterranean islands. In particular, a case study of a possible travel itinerary from Milan to the island of Lipari was analysed, also discussing a potential Smart Contract model. Blockchain is one of the technologies that could revolutionize the legal field, paving the way towards the innovative concept of "legal engineering". However, we believe that for its effective use we must overcome various ideological barriers erected above all by purists of traditional jurisprudence who do not look favorably on the prospect that the jurist's workflow could be distorted. This section opens a focus on the fact that the transition process towards the modernization and automation of jurisprudence could be long and not without difficulties. It is clear that the advantages deriving from Blockchain technology would bring undeniable advantages in the practice of drafting new laws, legal regulations and/or legal documents, facilitating the interpretation of existing ones and avoiding ambiguity. However, the transport sector itself could be the driving force of this technological transition by introducing new workflows to the advantage of stackholders.

CHAPTER 12

Conclusion and Highlights for Future Research

In this PhD thesis, innovative ICT technologies that can be used in the context of urban mobility have been discussed. The research questions were divided into two topics: Data Management and Service Management. The research questions relating to the Data Management topic were discussed in chapters 3, 4, 5 e 6. These chapters describe the studies carried out in the Big Data field concerning real use cases. The presented work considers aspects of data security, reliability, querying, and visualization. It was highlighted that the use of data even in the context of urban mobility suffers from security and reliability problems due to telecommunications technologies and protocols, computing units, and sensors. Solutions to the identified problems have been proposed. Furthermore, the problem of querying large databases was addressed. The need found is increasingly of practical use in smart cities. A solution was proposed that solved the identified problems and was then employed in a real use case in the context of decision support systems for urban mobility. The issue of data visualization was also addressed with reference to the geo-referencing of elements and visualization methods. Also in this case the solution identified was used in a real use case. The research questions relating to the Services Management topic were discussed in chapters 7, 8, 9, 10 e 11. In these chapters, reference is made to services that can be used in the context of urban mobility. From the user's point of view, it has been observed that in modern urban environments, citizens do not only need information on means of transport. Citizens use means of transport to reach shops or services, and for a better experience, they may need support inside buildings. Other times it could even be useful to have a service that does not make the citizen move from his home, this can occur in case of risk or simply in case of difficulty for the citizen himself. Technological solutions have been proposed to support citizens in the situations described and in particular in the case of the exploration of buildings. However, the study of services also covered the aspect relating to the devices that allow the collection of data and the management of automated systems. The aspect investigated concerned the optimization of the ability of a device to perform multiple functions. A system has been designed and tested for the automatic deployment of microservices which, starting from standardized protocols, optimizes the use and number of Edge devices to be installed in an urban environment. The current conditions of urban environments which are populated by numerous Edge/IoT devices were also analysed. The analysis of the problems has allowed us to design methods for optimizing the use of resources with federated systems also in artificial intelligence applications. Innovative applications have been described to support the diffusion of the concept of sustainable mobility in smart cities and to collect data directly from users. It has been demonstrated how Edge devices can also support the user from the point of view of personal safety while moving in an urban environment. These applications have been used in real scenarios. Finally, the use of Blockchain was evaluated as a possible technology to be used in advanced urban environments to support territorial continuity and traveler protection. This PhD thesis contains the results achieved to respond to the problems presented in the research project to which the activities referred. However, the technological evolution of urban environments and services provided to users always opens up new research scenarios. The solutions identified constitute a basis on which to work on future scenarios. The advent of artificial intelligence (AI), quantum computing, and autonomous vehicle driving services requires that the solutions identified are improved and adapted to the new future scenarios. However, the solutions described are valid and concepts such as IoT Rejuvenation and the security of Edge/IoT devices can find widespread use in use cases that are particularly relevant in urban mobility scenarios. On the other hand, applications for data visualization, support for the diffusion of sustainable mobility, and building navigation are today used in experimental research projects. The new needs will allow the works presented to evolve and make them become commonly used. Technicians, citizens, and researchers will increasingly use data as elements from which to derive value. It will therefore be necessary to improve the proposed solutions in order to make the data available in real-time, guaranteeing both high services and the economic and environmental sustainability of the technological applications. The introduction of technologies such as artificial intelligence, quantum computing and the advanced hardware capabilities of the IoT and Edge computing can radically transform the sector. AI could optimize traffic and facilitate the use of autonomous vehicles, while quantum algorithms could improve route planning and communications security. Advanced hardware capabilities, used for electric vehicles or advanced sensors, would help make urban transport more efficient and sustainable. The agile design of infrastructures and the adoption of open standards to propose the solutions described in this thesis are essential to allow the easy integration of new technologies. The intrinsic capacity for updating and transformation depends on the flexibility of the infrastructures and their ability to interact with different technologies and standards. In summary, in this PhD thesis, we addressed, from the ICT point of view, some of the major current urban mobility issues. In the future, more intelligent and flexible solutions are required, along with their integration with new emerging technologies, to improve the efficiency, safety and sustainability of transport in the urban area.

Bibliography

[1] Francesco Martella, Giovanni Parrino, Giuseppe Ciulla, Roberto Di Bernardo, Antonio Celesti, Maria Fazio, and Massimo Villari. Virtual device model extending ngsi-ld for faas at the edge. pages 660–667, 2021. (Cited at pages xv, 7, 211, 212, 236 e 243)

[2] Valeria Lukaj, Francesco Martella, Maria Fazio, Antonio Celesti, and Massimo Villari. Trusted ecosystem for iot service provisioning based on brokering. pages 746–753, 2021. (Cited at pages xv, 5, 82, 91, 137, 145, 236 e 272)

[3] Alessio Catalfamo, Maria Fazio, Francesco Martella, Antonio Celesti, and Massimo Villari. Muovime: Secure access to sustainable mobility services in smart city. pages 1–5, 2021. (Cited at pages xv e 8)

[4] Agata Romano, Rosaria Lanza, Fabrizio Celesti, Antonio Celesti, Maria Fazio, Francesco Martella, Antonino Galletta, and Massimo Villari. Towards smart tele-biomedical laboratory: Where we are, issues, and future challenges. In *2021 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–4, 2021. (Cited at pages xv e 6)

[5] Lorenzo Carnevale, Armando Ruggeri, Francesco Martella, Antonio Celesti, Maria Fazio, and Massimo Villari. Multi hop reconfiguration of end-devices in heterogeneous edge-iot mesh networks. In *2021 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6, 2021. (Cited at pages xv, 7 e 244)

[6] Antonino Quattrocchi, Damiano Alizzio, Francesco Martella, Valeria Lukaj, Massimo Villari, and Roberto Montanini. Effects of accelerated aging on the performance of

low-cost ultrasonic sensors used for public lighting and mobility management in smart cities. *Sensors*, 22(4), 2022. (Cited at pages xvi, 5, 107, 113, 115, 120, 212 e 272)

[7] Francesco Martella, Giovanni Parrino, Mario Colosi, Giuseppe Ciulla, Roberto Bernardo, Marco Martorana, Roberto Callari, Maria Fazio, Antonio Celesti, and Massimo Villari. Urbanite: Messina use case in smart mobility scenario. 10 2021. (Cited at pages xvi e 5)

[8] Giuseppe Ciulla, Roberto Bernardo, Isabel Matranga, Francesco Martella, and Giovanni Parrino. How disruptive technologies can strengthen urban mobility transformation. the experience of urbanite h2020 project shabnam farahmand. 10 2021. (Cited at pages xvi e 4)

[9] Mario Colosi, Francesco Martella, Giovanni Parrino, Antonio Celesti, Maria Fazio, and Massimo Villari. Time series data management optimized for smart city policy decision. In *2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, pages 585–594, 2022. (Cited at pages xvi e 4)

[10] Valeria Lukaj, Francesco Martella, Antonio Celesti, Maria Fazio, and Massimo Villari. An enriched visualization tool based on google maps for water distribution networks in smart cities. In *2022 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6, 2022. (Cited at pages xvi e 5)

[11] Francesco Martella, Maria Fazio, Antonio Celesti, Valeria Lukaj, Antonino Quattrocchi, Massimo Di Gangi, and Massimo Villari. Federated edge for tracking mobile targets on video surveillance streams in smart cities. In *2022 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6, 2022. (Cited at pages xvi, 7, 119 e 211)

[12] Valeria Lukaj, Francesco Martella, Antonino Quattrocchi, Maria Fazio, Roberto Montanini, Massimo Villari, and Antonio Celesti. Towards iot rejuvenation: a study on hy-srf05 ultrasonic sensor ageing for intelligent street pole lamp control in a smart city. In *2022 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–8, 2022. (Cited at pages xvi, 5, 91, 119, 212 e 272)

[13] Valeria Lukaj, Francesco Martella, Maria Fazio, Armando Ruggeri, Antonio Celesti, and Massimo Villari. A blockchain based federated ecosystem for tracking and validating the authenticity of goods. In *2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big*

Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), pages 1–7, 2022. (Cited at pages xvi e 8)

[14] Francesco Martella, Maria Fazio, Giuseppe Ciulla, Roberto Di Bernardo, Antonio Celesti, Valeria Lukaj, Mario Colosi, Massimo Di Gangi, and Massimo Villari. An edge system for the safety of cyclists in the urban area. In *2022 IEEE International Smart Cities Conference (ISC2)*, pages 1–7, 2022. (Cited at pages xvii e 8)

[15] Valeria Lukaj, Christian Sicari, Francesco Martella, Antonio Celesti, Maria Fazio, and Massimo Villari. An innovative method for 3d virtual indoor navigation based on geotags. 11 2022. (Cited at pages xvii e 6)

[16] Alessio Catalfamo, Lorenzo Carnevale, Antonino Galletta, Francesco Martella, Antonio Celesti, Maria Fazio, and Massimo Villari. Scaling data analysis services in an edge-based federated learning environment. In *2022 IEEE/ACM 15th International Conference on Utility and Cloud Computing (UCC)*, pages 167–172, 2022. (Cited at pages xvii e 7)

[17] Valeria Lukaj, Francesco Martella, Maria Fazio, Antonio Celesti, and Massimo Villari. Establishment of a trusted environment for iot service provisioning based on x3dh-based brokering and federated blockchain. *Internet of Things*, 21:100686, 2023. (Cited at pages xvii, 5, 82 e 91)

[18] Francesco Martella, Valeria Lukaj, Maria Fazio, Antonio Celesti, and Massimo Villari. On-demand and automatic deployment of microservice at the edge based on ngsi-ld. In *2023 31st Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, pages 314–320, 2023. (Cited at pages xvii e 7)

[19] Valeria Lukaj, Francesco Martella, Maria Fazio, Antonino Galletta, Antonio Celesti, and Massimo Villari. Gateway-based certification approach to include iot nodes in a trusted edge/cloud environment. 05 2023. (Cited at pages xvii e 5)

[20] Valeria Lukaj, Alessio Catalfamo, Francesco Martella, Maria Fazio, Massimo Villari, and Antonio Celesti. A nosql dbms transparent data encryption approach for cloud/edge continuum. 07 2023. (Cited at pages xvii e 5)

[21] Mehmet Sakman., Panagiotis Gkikopoulos., Francesco Martella., Massimo Villari., and Josef Spillner. Indoor navigation for personalised shopping: A real-tech feasibility study. In *Proceedings of the 20th International Conference on Smart Business Technologies - ICSBT*, pages 43–53. INSTICC, SciTePress, 2023. (Cited at pages xvii e 6)

[22] Roberto Montanini Antonio Quattrocchi Damiano Alizzi Francesco Martella Lukaj Valeria, Massimo Villari. Valutazione degli effetti dell'invecchiamento accelerato sulle performance di sensori low-cost di posizione impiegati in ambito smart cities, 2021. (Cited at pages xviii e 5)

[23] Antonino Quattrocchi, Francesco Martella, Valeria Lukaj, Rocco De Leo, Massimo Villari, and Roberto Montanini. Designing a low-cost system to monitor the structural behavior of street lighting poles in smart cities. *Sensors*, 23(15), 2023. (Cited at pages xviii e 5)

[24] Armando Ruggeri, Antonio Celesti, Valeria Lukaj, Francesco Martella, Ilenia Celesti, Maria Fazio, and Massimo Villari. Prospettive sull' utilizzo dello "smart contract" a supporto della continuità territoriale per la prenotazione dinamica di viaggi. In *LA CONTINUITÀ TERRITORIALE NEL TRASPORTO AEREO CON LE ISOLE DEL MEDITERRANEO*, 2023. (Cited at pages xviii e 8)

[25] Dmitry Namiot. Time series databases. 10 2015. (Cited at page 36)

[26] Roman Čerešňák and Michal Kvet. Comparison of query performance in relational a non-relation databases. *Transportation Research Procedia*, 40:170–177, 2019. TRANSCOM 2019 13th International Scientific Conference on Sustainable, Modern and Safe Transport. (Cited at page 36)

[27] Mahmoudreza Tahmassebpour. A new method for time-series big data effective storage. *IEEE Access*, 5:10694–10699, 2017. (Cited at page 36)

[28] Souad Amghar, Safae Cherdal, and Salma Mouline. Storing, preprocessing and analyzing tweets: Finding the suitable nosql system, 05 2020. (Cited at page 37)

[29] Muon Ha and Yulia Shichkina. The query translation from mysql to mongodb taking into account the structure of the database. In *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*, pages 383–386, 2021. (Cited at page 37)

[30] Sergio Di Martino, Luca Fiadone, Adriano Peron, Alberto Riccabone, and Vincenzo Norman Vitale. Industrial internet of things: Persistence for time series with nosql databases. In *2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pages 340–345, 2019. (Cited at page 37)

[31] Eugene Siow, Thanassis Tiropanis, Xin Wang, and Wendy Hall. Tritandb: Time-series rapid internet of things analytics. *CoRR*, abs/1801.07947, 2018. (Cited at page 37)

[32] Denis Arnst, Valentin Plenk, and Adrian Wöltche. Comparative evaluation of database performance in an internet of things context. 10 2018. (Cited at page 37)

[33] Stefan Jovanov, Vladimir Zdraveski, and Marjan Gusev. Gpu in applications with non-relational dbs. In *2020 28th Telecommunications Forum (TELFOR)*, pages 1–4, 2020. (Cited at page 37)

[34] Xiao Mo and Hao Wang. Asynchronous index strategy for high performance real-time big data stream storage. In *2012 3rd IEEE International Conference on Network Infrastructure and Digital Content*, pages 232–236, 2012. (Cited at page 38)

[35] Peeyush Gupta, Michael J. Carey, Sharad Mehrotra, and oberto Yus. Smartbench: A benchmark for data management in smart spaces. *Proc. VLDB Endow.*, 13(12):1807–1820, jul 2020. (Cited at page 38)

[36] Silvia Comunian, Dario Dongo, Chiara Milani, and Paola Palestini. Air pollution and covid-19: The role of particulate matter in the spread and increase of covid-19's morbidity and mortality. *International Journal of Environmental Research and Public Health*, 17(12), 2020. (Cited at page 56)

[37] Chiara Copat, Antonio Cristaldi, Maria Fiore, Alfina Grasso, Pietro Zuccarello, Santo Signorelli, Gea Conti, and Margherita Ferrante. The role of air pollution (pm and no2) in covid-19 spread and lethality: A systematic review. *Environmental Research*, 191:110129, 08 2020. (Cited at page 56)

[38] Alina Machidon, Maj Smerkol, and Matjaž Gams. Urbanite h2020 project. algorithms and simulation techniques for decision – makers. *Proceedings of the 23rd International Multiconference INFORMATION SOCIETY*, A:68–71, 2020. (Cited at page 56)

[39] Maj Smerkol, Žan Počkar, Alina Machidon, and Matjaž Gams. Traffic simulation software in the context of mobility policy support system. *In Information Society 2020*, 2020. (Cited at page 56)

[40] *CYCLING MATTERS 2019, How Bicycles Power Amsterdam*. CITY OF AMSTERDAM, may 2019. (Cited at page 57)

[41] *Plan de Movilidad Urbana Sostenible (PMUS) 2015-2030 de la Villa de Bilbao, Fase II. Propuesta*. Ayuntamiento de Bilbao, Área de Movilidad y Sostenibilidad, may 2018. (Cited at page 58)

[42] Alessio Catalfamo, Maria Fazio, Francesco Martella, Antonio Celesti, and Massimo Villari. MuoviMe: secure access to sustainable mobility services in smart city. September 2021. (Cited at pages 61, 136 e 272)

[43] Francesco Martella, Giovanni Parrino, Giuseppe Ciulla, Roberto Di Bernardo, Antonio Celesti, Maria Fazio, and Massimo Villari. Virtual device model extending ngsi-ld for faas at the edge. In *2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, pages 660–667, 2021. (Cited at pages 61, 137 e 273)

[44] Lorenzo Carnevale, Antonio Celesti, Maria Di Pietro, and Antonino Galletta. How to conceive future mobility services in smart cities according to the fiware frontiercities experience. *IEEE Cloud Computing*, 5(5):25–36, 2018. (Cited at pages 61 e 136)

[45] E. Kim, K. Chung, and T. Jeong. Self-certifying id based trustworthy networking system for iot smart service domain. In *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1299–1301, 2017. (Cited at pages 66 e 82)

[46] T. Shah and S. Venkatesan. Authentication of iot device and iot server using secure vaults. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 819–824, 2018. (Cited at page 66)

[47] M. Togan, B. Chifor, I. Florea, and G. Gugulea. A smart-phone based privacy-preserving security framework for iot devices. In *2017 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pages 1–7, 2017. (Cited at page 66)

[48] P. Urien. An innovative security architecture for low cost low power iot devices based on secure elements: A four quarters security architecture. In *2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC)*, pages 1–2, 2018. (Cited at page 66)

[49] T. Marktscheffel, W. Gottschlich, W. Popp, P. Werli, S. D. Fink, A. Bilzhause, and H. de Meer. Qr code based mutual authentication protocol for internet of things. In *2016*

IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), pages 1–6, 2016. (Cited at page 66)

[50] I. Chen, J. Guo, D. Wang, J. J. P. Tsai, H. Al-Hamadi, and I. You. Trust as a service for iot service management in smart cities. In *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pages 1358–1365, 2018. (Cited at pages 66 e 82)

[51] T. Akeem Yekini, F. Jaafar, and P. Zavarsky. Study of trust at device level of the internet of things architecture. In *2019 IEEE 19th International Symposium on High Assurance Systems Engineering (HASE)*, pages 150–155, 2019. (Cited at page 66)

[52] J. Thakker, I. Chang, and Y. Park. Secure data management in internet-of-things based on blockchain. In *2020 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–5, 2020. (Cited at page 67)

[53] R. T. Frahat, M. M. Monowar, and S. M. Buhari. Secure and scalable trust management model for iot p2p network. In *2019 2nd International Conference on Computer Applications Information Security (ICCAIS)*, pages 1–6, 2019. (Cited at pages 67 e 82)

[54] M. Pahl and L. Donini. Securing iot microservices with certificates. In *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–5, 2018. (Cited at pages 67 e 82)

[55] M. W. Condry and C. B. Nelson. Using smart edge iot devices for safer, rapid response with industry iot control operations. *Proceedings of the IEEE*, 104(5):938–946, 2016. (Cited at page 67)

[56] G. Dittmann and J. Jelitto. A blockchain proxy for lightweight iot devices. In *2019 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pages 82–85, 2019. (Cited at page 67)

[57] Davide Mulfari, Gabriele Meoni, Marco Marini, and Luca Fanucci. Machine learning assistive application for users with speech disorders. *Applied Soft Computing*, 103:107147, 2021. (Cited at page 68)

[58] R. Shantha Joshitta. Security in iot environment: A survey. *Int. Journal of Information Technology & Mechanical Engineering*, 2:1–8, 07 2016. (Cited at page 69)

[59] L. Barreto, A. Celesti, M. Villari, M. Fazio, and A. Puliafito. Identity management in iot clouds: A fiware case of study. In *2015 IEEE Conference on Communications and Network Security (CNS)*, pages 680–684, 2015. (Cited at page 71)

[60] Global iot market to grow to 24.1 billion devices in 2030, generating $1.5 trillion annual revenue. `https://transformainsights.com/news/iot-market-24-billion-usd15-trillion-revenue-2030`. Accessed: 2023-02-08. (Cited at page 80)

[61] Zhipeng Liu, Niraj Thapa, Addison Shaver, Kaushik Roy, Xiaohong Yuan, and Sajad Khorsandroo. Anomaly detection on iot network intrusion using machine learning. In *2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, pages 1–5, 2020. (Cited at page 80)

[62] Elisa Bertino and Nayeem Islam. Botnets and internet of things security. *Computer*, 50(2):76–79, 2017. (Cited at page 80)

[63] Ruchi Vishwakarma and Ankit Kumar Jain. A honeypot with machine learning based detection framework for defending iot based botnet ddos attacks. In *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 1019–1024, 2019. (Cited at page 80)

[64] Muhammad Shafiq, Zhihong Tian, Ali Kashif Bashir, Xiaojiang Du, and Mohsen Guizani. Corrauc: A malicious bot-iot traffic detection method in iot network using machine-learning techniques. *IEEE Internet of Things Journal*, 8(5):3242–3254, 2021. (Cited at page 80)

[65] Yair Meidan, Michael Bohadana, Yael Mathov, Yisroel Mirsky, Asaf Shabtai, Dominik Breitenbacher, and Yuval Elovici. N-baiot—network-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Computing*, 17(3):12–22, 2018. (Cited at page 80)

[66] Mohammed Amine Bouras, Qinghua Lu, Sahraoui Dhelim, and Huansheng Ning. A lightweight blockchain-based iot identity management approach. *Future Internet*, 13(2), 2021. (Cited at page 80)

[67] Sara N. Matheu-García, José L. Hernández-Ramos, Antonio F. Skarmeta, and Gianmarco Baldini. Risk-based automated assessment and testing for the cybersecurity

certification and labelling of iot devices. *Computer Standards & Interfaces*, 62:64–83, 2019. (Cited at page 81)

[68] G. Baldini, A. Skarmeta, E. Fourneret, R. Neisse, B. Legeard, and F. Le Gall. Security certification and labelling in internet of things. In *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pages 627–632, 2016. (Cited at page 82)

[69] Juan Benet. Ipfs - content addressed, versioned, p2p file system. 07 2014. (Cited at page 82)

[70] R. Neisse, J. L. Hernández-Ramos, S. N. Matheu, G. Baldini, and A. Skarmeta. Toward a blockchain-based platform to manage cybersecurity certification of iot devices. In *2019 IEEE Conference on Standards for Communications and Networking (CSCN)*, pages 1–6, 2019. (Cited at page 82)

[71] Daniel Minoli and Benedict Occhiogrosso. Blockchain mechanisms for iot security. *Internet of Things*, 1-2:1–13, 2018. (Cited at page 82)

[72] O. B. Mora, R. Rivera, V. M. Larios, J. R. Beltrán-Ramírez, R. Maciel, and A. Ochoa. A use case in cybersecurity based in blockchain to deal with the security and privacy of citizens and smart cities cyberinfrastructures. In *2018 IEEE International Smart Cities Conference (ISC2)*, pages 1–4, 2018. (Cited at page 82)

[73] Phillip Williams, Indira Kaylan Dutta, Hisham Daoud, and Magdy Bayoumi. A survey on security in internet of things with a focus on the impact of emerging technologies. *Internet of Things*, 19:100564, 2022. (Cited at page 83)

[74] Sara N. Matheu, José L. Hernández-Ramos, Antonio F. Skarmeta, and Gianmarco Baldini. A survey of cybersecurity certification for the internet of things. 53(6), dec 2020. (Cited at page 83)

[75] Ganache ethereum simulation. `https://trufflesuite.com/ganache/`. Accessed: 2023-01-25. (Cited at page 86)

[76] Daniel Maniglia Amancio Da Silva and Rute C. Sofia. A discussion on context-awareness to better support the iot cloud/edge continuum. *IEEE Access*, 8:193686–193694, 2020. (Cited at page 90)

[77] Elahe Fazeldehkordi and Tor-Morten Grønli. A survey of security architectures for edge computing-based iot. *IoT*, 3(3):332–365, 2022. (Cited at page 91)

[78] Kewei Sha, T. Andrew Yang, Wei Wei, and Sadegh Davari. A survey of edge computing-based designs for iot security. *Digital Communications and Networks*, 6(2):195–202, 2020. (Cited at page 91)

[79] Qian Liu, Juan Gu, Jingchao Yang, Yun Li, Dexuan Sha, Mengchao Xu, Ishan Shams, Manzhu Yu, and Chaowei Yang. *Cloud, Edge, and Mobile Computing for Smart Cities*, pages 757–795. Springer Singapore, Singapore, 2021. (Cited at page 91)

[80] Evaristus Didik Madyatmadja, Aditya Nur Hakim, and David Jumpa Malem Sembiring. Performance testing on transparent data encryption for sql server's reliability and efficiency. *Journal of Big Data*, 8(1):134, 2021. (Cited at page 91)

[81] Dr. Anwar Pasha Deshmukh and Dr. Riyazuddin Qureshi. Transparent data encryption – solution for security of database contents. 2013. (Cited at page 91)

[82] K Natarajan and Vaheedbasha Shaik. Transparent data encryption: Comparative analysis and performance evaluation of oracle databases. In *2020 Fifth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, pages 137–142, 2020. (Cited at page 91)

[83] N. Sai Sirisha and K. Venkat Kiran. Authorization of data in hadoop using apache sentry. *International Journal of Engineering & Technology*, 2018. (Cited at page 91)

[84] Devanshu Trivedi, Pavol Zavarsky, and Sergey Butakov. Enhancing relational database security by metadata segregation. *Procedia Computer Science*, 94:453–458, 2016. The 11th International Conference on Future Networks and Communications (FNC 2016) / The 13th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2016) / Affiliated Workshops. (Cited at page 91)

[85] Vasily Sidorov and Wee Keong Ng. Transparent data encryption for data-in-use and data-at-rest in a cloud-based database-as-a-service solution. In *2015 IEEE World Congress on Services*, pages 221–228, 2015. (Cited at page 92)

[86] Joao Sousa, Tiago Mateus, and Pedro Furtado. Scalability architecture for a secure big data system. *International Journal of Business Process Integration and Management*, 7:345, 01 2015. (Cited at page 92)

[87] Morris Dworkin, Elaine Barker, James Nechvatal, James Foti, Lawrence Bassham, E. Roback, and James Dray. Advanced encryption standard (aes), 2001-11-26 2001. (Cited at page 92)

[88] Na Su, Yi Zhang, and Mingyue Li. Research on data encryption standard based on aes algorithm in internet of things environment. In *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pages 2071–2075, 2019. (Cited at page 92)

[89] Pedro Sanchez Munoz, Nam Tran, Brandon Craig, Behnam Dezfouli, and Yuhong Liu. Analyzing the resource utilization of aes encryption on iot devices. In *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1200–1207, 2018. (Cited at page 92)

[90] Pratibha Vuppuluri. Investing in innovation: The rise of the smart city. `https://www.forbes.com/sites/forbesfinancecouncil/2020/12/03/investing-in-innovation-the-rise-of-the-smart-city/?sh=76b71b4c5ba6`, December 2020. (Cited at page 103)

[91] Knud Lasse Lueth. The impact of covid-19 on the internet of things – now and beyond the great. `https://iot-analytics.com/the-impact-of-covid-19-on-the-internet-of-things-part-2/?utm_source=Covid%20Landing%20Page&utm_medium=ButtonBox1&utm_campaign=Covid%20Campaign`, April 2020. (Cited at page 103)

[92] T. Lynn, P. Rosati, A. Lejeune, and V. Emeakaroha. A preliminary review of enterprise serverless cloud computing (function-as-a-service) platforms. In *2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 162–169, 2017. (Cited at page 105)

[93] T. Pfandzelter and D. Bermbach. tinyfaas: A lightweight faas platform for edge environments. In *2020 IEEE International Conference on Fog Computing (ICFC)*, pages 17–24, 2020. (Cited at page 105)

[94] R Nikhil, B S Anisha, and P Ramakanth Kumar. Real-time monitoring of agricultural land with crop prediction and animal intrusion prevention using internet of things and machine learning at edge. 2020. (Cited at page 105)

[95] M. A. Romli, S. Daud, S. M. Zainol, P. L. E. Kan, and Z. A. Ahmad. Automatic ras data acquisition and processing system using fog computing. In *2017 IEEE 13th Malaysia International Conference on Communications (MICC)*, pages 229–234, 2017. (Cited at page 105)

[96] Zheng Gong, Wuyang Xue, Ziang Liu, Yimo Zhao, Ruihang Miao, Rendong Ying, and Peilin Liu. Design of a reconfigurable multi-sensor testbed for autonomous vehicles and ground robots. pages 1–5, 05 2019. (Cited at page 106)

[97] Anif Jamaluddin, Fita Listiana, Dwi Rahardjo, Lita Rahmasari, and Dewanto Harjunowibowo. Simple method for non contact thickness gauge using ultrasonic sensor and android smartphone. *TELKOMNIKA Indonesian Journal of Electrical Engineering*, 15, 07 2015. (Cited at page 106)

[98] Nuryanto, A. Widiyanto, and A. Burhanuddin. Redirection concept of autonomous mobile robot hy-srf05 sensor to reduce the number of sensors. In *2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, pages 1–4, 2017. (Cited at page 106)

[99] V A Zhmud, N O Kondratiev, K A Kuznetsov, V G Trubin, and L V Dimitrov. Application of ultrasonic sensor for measuring distances in robotics. *Journal of Physics: Conference Series*, 1015:032189, may 2018. (Cited at page 106)

[100] P. Haguenauer, E. Fedrigo, L. Pettazzi, C. Reinero, F. Gonte, L. Pallanca, R. Frahm, J. Woillez, and P. Lilley. Rejuvenation of a ten-year old ao curvature sensor: Combining obsolescence correction and performance upgrade of macao. volume 9909, 2016. (Cited at page 106)

[101] M. Woehrle, A. Meier, and K. Langendoen. On the potential of software rejuvenation for long-running sensor network deployments. pages 44–48, 2010. (Cited at page 106)

[102] S. Parvin, D.S. Kim, S.M. Lee, and J.S. Park. Achieving availability and survivability in wireless sensor networks by software rejuvenation. pages 13–18, 2008. (Cited at page 106)

[103] D.S. Kim, S. Parvin, and J.S. Park. Software rejuvenation and reconfiguration for enhancing survivability of sensor networks. pages 732–737, 2008. (Cited at page 107)

[104] S. Parvin, D.S. Kim, and J.S. Park. Towards optimal software rejuvenation in wireless sensor networks using self-regenerative components. pages 295–300, 2008. (Cited at page 107)

[105] S. Parvin, D.S. Kim, and J.S. Park. Towards survivable sensor networks using self-regenerative rejuvenation and reconfiguration. pages 546–549, 2007. (Cited at page 107)

[106] Francisco Airton Silva, Carlos Brito, Gabriel Araújo, Iure Fé, Maxim Tyan, Jae-Woo Lee, Tuan Anh Nguyen, and Paulo Romero Martin Maciel. Model-driven impact quantification of energy resource redundancy and server rejuvenation on the dependability of medical sensor networks in smart hospitals. *Sensors*, 22(4), 2022. (Cited at page 107)

[107] Konrad Henryk Bachanek, Blanka Tundys, Tomasz Wiõniewski, Ewa Puzio, and Anna Maroukov. Intelligent street lighting in a smart city concepts,Äîa direction to energy saving in cities: An overview and case study. *Energies*, 14(11), 2021. (Cited at page 119)

[108] Omar A.; AlMaeeni S.; Sanduleanu M.; Shubair R.; Ashhab M.S.; Al Ali M.; Al Hebsi G. Smart city: Recent advances in intelligent street lighting systems based on iot. *Sens*, 249187S., 2022. (Cited at page 119)

[109] Syed Najeeb Ali Kazmi, Abasin Ulasyar, and Faisal Nadeem. Iot based energy efficient smart street lighting technique with air quality monitoring. pages 1–6, 12 2020. (Cited at page 119)

[110] Francesco Marino, Fabio Leccese, and Stefano Pizzuti. Adaptive street lighting predictive control. *Energy Procedia*, 111:790–799, 2017. 8th International Conference on Sustainability in Energy and Buildings, SEB-16, 11-13 September 2016, Turin, Italy. (Cited at page 119)

[111] Ashish Pandharipande and Paul Thijssen. Connected street lighting infrastructure for smart city applications. *IEEE Internet of Things Magazine*, 2:32–36, 06 2019. (Cited at page 119)

[112] Giuseppe Parise, Luigi Martirano, and Massimo Mitolo. Electrical safety of street light systems. *Power Delivery, IEEE Transactions on*, 26:1952 – 1959, 08 2011. (Cited at page 119)

[113] Zulkifli Ishak, Wan Siti Halimatul Munirah Wan Ahmad, Nurul Asyikin Mohamed Radzi, Suhaila Sulaiman, and Noor Emilia Ramli. Placement accuracy algorithm for smart street lights. *Turkish J. Electr. Eng. Comput. Sci.*, 29(2):845–859, 2021. (Cited at page 119)

[114] Eric Merschman, Abdullahi M Salman, Emilio Bastidas-Arteaga, and Yue Li. Assessment of the effectiveness of wood pole repair using frp considering the impact of climate change on decay and hurricane risk. *Advances in Climate Change Research*, 11(4):332–348, 2020. (Cited at page 119)

[115] Thomas E Rodgers Jr. Prestressed concrete poles: state-of-the-art. *PCI Journal*, 29(5):52–103, 1984. (Cited at page 119)

[116] T Hamouda. Complex three-dimensional-shaped knitting preforms for composite application. *Journal of Industrial Textiles*, 46(7):1536–1551, 2017. (Cited at page 119)

[117] Ze Feng Zheng and Ke Su. The design of landscape lighting based on wind, light, water and other green energy. *Applied Mechanics and Materials*, 686:643–647, 2014. (Cited at page 119)

[118] Nan Su Mon Aung and Zaw Htet Myint. Design of stand-alone solar street lighting system with led. *International journal of scientific engineering and technology research*, 3(17):3518–3522, 2014. (Cited at page 119)

[119] Åke Wisten. Impact of lightning on street lights:-an experimental study investigating different poles and cables. In *2020 International Symposium on Electromagnetic Compatibility-EMC EUROPE*, pages 1–7. IEEE, 2020. (Cited at page 119)

[120] Sun Hee Kim, Gi Nam Kim, Soon Jung Hong, Chang Won Kim, Won Sup Jang, and Soon Jong Yoon. Structural behavior of frp lighting pole system. In *Materials Science Forum*, volume 654, pages 1034–1037. Trans Tech Publ, 2010. (Cited at page 119)

[121] Goutam Das, S Chakrabarty, AK Dutta, Swapan K Das, KK Gupta, and RN Ghosh. Failure analysis of a high mast lamp post. *Engineering Failure Analysis*, 13(7):1153–1158, 2006. (Cited at page 119)

[122] Reza Nasouri, Kien Nguyen, Arturo Montoya, Adolfo Matamoros, Caroline Bennett, and Jian Li. Simulating the hot dip galvanizing process of high mast illumination poles. part ii: Effects of geometrical properties and galvanizing practices. *Journal of Constructional Steel Research*, 159:584–597, 2019. (Cited at page 120)

[123] Ahmed Elmarakbi, Niki Fielding, and Homayoun Hadavinia. Finite element simulation of axial crush of thin-walled tubes with different cross-sections: vehicle/pole impact applications. *International Journal of Vehicle Structures & Systems*, 3(3):154, 2011. (Cited at page 120)

[124] IO Mockey Coureaux and E Manzano. The energy impact of luminaire depreciation on urban lighting. *energy for sustainable development*, 17(4):357–362, 2013. (Cited at page 120)

[125] Suma Devi and L Govindaraju. A study on wind induced vibration on lighting poles. *IJRET: International Journal of Research in Engineering and Technology*, 3(06), 2014. (Cited at page 120)

[126] Paraic C Ryan, Mark G Stewart, Nathan Spencer, and Yue Li. Reliability assessment of power pole infrastructure incorporating deterioration and network maintenance. *Reliability Engineering & System Safety*, 132:261–273, 2014. (Cited at page 120)

[127] A Howard. Non-destructive testing of lighting columns: an assessment of current and future testing methods. *Lighting Journal*, 63(6), 1998. (Cited at page 120)

[128] Marcin Ziolkowski, Rajesh Taneja, Didar S Dulay, M Imran Rafiq, and Dhaivat Jani. Development of novel" short range ultrasonic guided waves"(srugw) technique for lighting poles inspection. In *BINDT 2007 Conference, 18Y20 September*, 2007. (Cited at page 120)

[129] Bjørn Gustavsen and Lars Rolfseng. Asset management of wood pole utility structures. *International Journal of Electrical Power & Energy Systems*, 27(9-10):641–646, 2005. (Cited at page 120)

[130] Margarit G Lozev et al. Prototype crawling robotics system for remote visual inspection of high-mast light poles. Technical report, Virginia Transportation Research Council, 1997. (Cited at page 120)

[131] Pavel Steinbauer, Zdenek Neusser, Ivo Bukovsky, and Milos Neruda. Lighting pole health monitoring for predictive maintenance. *Procedia Structural Integrity*, 17:799–805, 2019. (Cited at pages 120 e 122)

[132] Telecontrollo quadro - impianti di illuminazione pubblica, 2023. `https://www.revetec.it/en/products-category/4/telecontrollo-quadro` [Accessed: (08 April 2023)]. (Cited at page 120)

[133] Dht22 temperature and humidity sensor, 2023. `https://www.az-delivery.de/en/products/dht22` [Accessed: (08 April 2023)]. (Cited at page 121)

[134] Gy-302 bh1750 light sensor brightness sensor., 2023. `https://www.az-delivery.de/en/products` [Accessed: (08 April 2023)]. (Cited at page 121)

[135] Mpu-6050 3-axis gyroscope and acceleration sensor, 2023. `https://www.az-delivery.de/en/products` [Accessed: (08 April 2023)]. (Cited at page 122)

[136] Hayes E Ross Jr and Thomas C Edwards. Wind induced vibration in light poles. *Journal of the Structural Division*, 96(6):1221–1235, 1970. (Cited at page 122)

[137] Leena Tähkämö and Liisa Halonen. Life cycle assessment of road lighting luminaires–comparison of light-emitting diode and high-pressure sodium technologies. *Journal of Cleaner Production*, 93:234–242, 2015. (Cited at page 123)

[138] Rohaida M Ramli, Yanuar Z Arief, and Pusparini Dewi Abd Aziz. Application of led technology into public road lighting in malaysia for replacing the high pressure sodium vapour lighting. In *2015 International Conference on Sustainable Energy Engineering and Application (ICSEEA)*, pages 76–81. IEEE, 2015. (Cited at page 123)

[139] Armin Dadras Eslamlou, Aliakbar Ghaderiaram, Erik Schlangen, and Mohammad Fotouhi. A review on non-destructive evaluation of construction materials and structures using magnetic sensors. *Construction and Building Materials*, 397:132460, 2023. (Cited at page 123)

[140] Thunder scientific model 2500 humidity generator., 2023. `https://www.thunderscientific.com/model_2500/` [Accessed: (08 April 2023)]. (Cited at page 123)

[141] Luxmeter testo 540., 2023. `https://www.testo.com/it-IT/testo-540/p/0560-0540` [Accessed: (08 April 2023)]. (Cited at page 124)

[142] Schwingtechnik—tira gmbh. available, 2023. `https://www.tira-gmbh.dk/images/PDF/TIRA-selection-guide-eng-05-2015.pdf` [Accessed: (08 April 2023)]. (Cited at page 124)

[143] 33220a function/arbitrary waveform generator, 2023. `https://www.keysight.com/us/en/product/33220A/function{-}{-}arbitrary-waveform-generator-20-mhz.html` [Accessed: (08 April 2023)]. (Cited at page 124)

[144] Model 356a19—pcb piezotronics, 2023. `https://www.pcb.com/products?m=356a19` [Accessed: (08 April 2023)]. (Cited at page 124)

[145] Chassis ni compactdaq, 2023. `https://www.ni.com/it-it/shop/category/compactdaq-chassis.html?productId=118191` [Accessed: (08 April 2023)]. (Cited at page 124)

[146] Gam 270 mfl angle measurer, 2023. https://www.bosch-professional.com/pk/en/products/gam-270-mfl-0601076400 [Accessed: (08 April 2023)]. (Cited at page 124)

[147] Gam 270 mfl angle measurer, 2023. https://www.bosch-professional.com/pk/en/products/gam-270-mfl-0601076400 [Accessed: (08 April 2023)]. (Cited at pages 124 e 130)

[148] Antonino Galletta, Armando Ruggeri, Maria Fazio, Gianluca Dini, and Massimo Villari. Mesmart-pro: Advanced processing at the edge for smart urban monitoring and reconfigurable services. *Journal of Sensor and Actuator Networks*, 9(4):55, Dec 2020. (Cited at pages 137, 196 e 273)

[149] Luca Pappalardo, Filippo Simini, Gianni Barlacchi, and Roberto Pellungrini. scikit-mobility: a python library for the analysis, generation and risk assessment of mobility data, 2019. (Cited at pages 137 e 142)

[150] Marta C. Gonzalez, Cesar Hidalgo, and Albert-Laszlo Barabasi. Understanding individual human mobility patterns. *Nature*, 453:779–82, 07 2008. (Cited at page 137)

[151] Yves-Alexandre Montjoye, Cesar Hidalgo, Michel Verleysen, and Vincent Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports*, 3:1376, 03 2013. (Cited at page 137)

[152] Fiware4Water. https://www.fiware4water.eu/. (Cited at page 143)

[153] digital-water.city. https://www.digital-water.city/. (Cited at page 143)

[154] Gedela Venkata Ramana, Sudheer Chekka, and Rajasekhar Bellapu. Network analysis of water distribution system in rural areas using epanet. *Procedia Engineering*, 119:496–505, 12 2015. (Cited at page 144)

[155] R. Chalh, Z. Bakkoury, D. Ouazar, and M. D. Hasnaoui. Big data open platform for water resources management. In *2015 International Conference on Cloud Technologies and Applications (CloudTech)*, pages 1–8, 2015. (Cited at page 144)

[156] Jonathan Pickus, Rakesh Bahadur, and William Samuels. Integrating the arcgis water distribution data model into pipelinenet. 01 2005. (Cited at page 144)

[157] R. Ahmadullah and K. Dongshik. Construction of hydraulically balanced water distribution network. In *2015 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, pages 96–98, 2015. (Cited at page 144)

[158] A. Afaneh and I. Shahrour. Use of gis for sunrise smart city project, large scale demonstrator of the smart city. In *2017 Sensors Networks Smart and Emerging Technologies (SENSET)*, pages 1–4, 2017. (Cited at page 144)

[159] A. E. U. Salam, M. Tola, M. Selintung, and F. Maricar. A leakage detection system on the water pipe network through support vector machine method. In *2014 Makassar International Conference on Electrical Engineering and Informatics (MICEEI)*, pages 161–165, 2014. (Cited at page 144)

[160] J. Kemba, K. Gideon, and C. N. Nyirenda. Leakage detection in tsumeb east water distribution network using epanet and support vector regression. In *2017 IST-Africa Week Conference (IST-Africa)*, pages 1–8, 2017. (Cited at page 144)

[161] Christophe Dumora, David Auber, Jérémie Bigot, Vincent Couallier, and C. Leclerc. Data-oriented algorithm for real-time estimation of flow rates and flow directions in a water distribution network, 07 2018. (Cited at pages 144 e 150)

[162] Armando Di Nardo, Carlo Giudicianni, Roberto Greco, Manuel Herrera, Giovanni Francesco Santonastaso, and Antonio Scala. Sensor placement in water distribution networks based on spectral algorithms. In Goffredo La Loggia, Gabriele Freni, Valeria Puleo, and Mauro De Marchis, editors, *HIC 2018. 13th International Conference on Hydroinformatics*, volume 3 of *EPiC Series in Engineering*, pages 593–600. EasyChair, 2018. (Cited at page 144)

[163] Orazio Giustolisi, Antonietta Simone, and Luca Ridolfi. Network structure classification and features of water distribution systems. *Water Resources Research*, 53(4):3407–3423, 2017. (Cited at page 144)

[164] Tilmann Rabl, Christian Dellwo, and Harald Kosch. Introducing scalileo: A java based scaling framework. pages 205–214, 01 2010. (Cited at page 144)

[165] S. K. Alshattnawi. Smart water distribution management system architecture based on internet of things and cloud computing. In *2017 International Conference on New Trends in Computing Sciences (ICTCS)*, pages 289–294, 2017. (Cited at page 144)

[166] S. Shihu, Z. Dong, L. Suiqing, M. Mingqun, Z. Ming, Y. Yixing, G. Jinliang, and Z. Hongbin. Decision support system of water distribution network expansion. In *2010 International Conference on E-Business and E-Government*, pages 1520–1523, 2010. (Cited at page 144)

[167] Water Scarcity WWF. `https://www.worldwildlife.org/threats/water-scarcity`. (Cited at page 145)

[168] Marike Kellermayr-Scheucher, Laura Hörandner, and Patrick Brandtner. Digitalization at the point-of-sale in grocery retail - state of the art of smart shelf technology and application scenarios. *Procedia Computer Science*, 196:77–84, 2022. International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies 2021. (Cited at page 157)

[169] Taeshik Gong, Chen-Ya Wang, and Kangcheol Lee. Effects of characteristics of in-store retail technology on customer citizenship behavior. *Journal of Retailing and Consumer Services*, 65:102488, 2022. (Cited at page 157)

[170] Peter Linzbach, J. Jeffrey Inman, and Hristina Nikolova. E-commerce in a physical store: Which retailing technologies add real value? *NIM Marketing Intelligence Review*, 11(1):42–47, 2019. (Cited at page 157)

[171] Patricia Sanchez Miralles, Laura Fernandez Gonzalez, Xinrui Yu, and Jafar Saniie. Assisting visually impaired people using autonomous navigation system and computer vision for grocery shopping. In *2022 IEEE International Conference on Electro Information Technology, EIT 2022, Mankato, MN, USA, May 19-21, 2022*, pages 203–208. IEEE, 2022. (Cited at page 158)

[172] Shubham Deshmukh, Favin Fernandes, Amey Chavan, Monali Ahire, Devashri Borse, and Jyoti Madake. SANIP: shopping assistant and navigation for the visually impaired. *CoRR*, abs/2209.03570, 2022. (Cited at page 158)

[173] G. S. T. Perera, K. W. R. Madhubhashini, Dilani Lunugalage, D. V. S. Piyathilaka, W. H. U. Lakshani, and Dharshana Kasthurirathna. Computer vision based indoor navigation for shopping complexes. In *ICVISP 2020: 4th International Conference on Vision, Image and Signal Processing, Bangkok, Thailand, December, 2020*, pages 15:1–15:6. ACM, 2020. (Cited at page 158)

[174] Philip Ruijgrok, Flavius Frasincar, Damir Vandic, and Frederik Hogenboom. Ontonavshop: an ontology-based approach for web-shop navigation. *Journal of Web Engineering*, pages 241–269, 2018. (Cited at page 158)

[175] Ruan Heyns, Musa Ndiaye, and Adnan M. Abu-Mahfouz. Enabling user-oriented features at the edge: A case of an iot-based smart shopping cart. In *30th IEEE International Symposium on Industrial Electronics, ISIE 2021, Kyoto, Japan, June 20-23, 2021*, pages 1–6. IEEE, 2021. (Cited at page 158)

[176] Anne-Sophie Riegger, Jan F. Klein, Katrin Merfeld, and Sven Henkel. Technology-enabled personalization in retail stores: Understanding drivers and barriers. *Journal of Business Research*, 123:140–155, 2021. (Cited at page 158)

[177] Jengchung Victor Chen, Sirapattra Ruangsri, Quang-An Ha, and Andree E. Widjaja. An experimental study of consumers' impulse buying behaviour in augmented reality mobile shopping apps. *Behav. Inf. Technol.*, 41(15):3360–3381, 2022. (Cited at page 158)

[178] Chia-Chen Chen, Tien-Chi Huang, James J. Park, Huang-Hua Tseng, and Neil Y. Yen. A smart assistant toward product-awareness shopping. *Pers. Ubiquitous Comput.*, 18(2):339–349, 2014. (Cited at page 158)

[179] Joe Boden, Erik Maier, and Florian Dost. The effect of electronic shelf labels on store revenue. *International Journal of Electronic Commerce*, 24(4):527–550, 2020. (Cited at page 158)

[180] Marion Garaus, Elisabeth Wolfsteiner, and Udo Wagner. Shoppers' acceptance and perceptions of electronic shelf labels. *Journal of Business Research*, 69(9):3687–3692, 2016. (Cited at page 158)

[181] Quentin Vey, Réjane Dalcé, Adrien van den Bossche, and Thierry Val. Indoor UWB localisation: Locura4iot testbed and dataset presentation. In Sharief Oteafy, Eyuphan Bulut, and Florian Tschorsch, editors, *47th IEEE Conference on Local Computer Networks, LCN 2022, Edmonton, AB, Canada, September 26-29, 2022*, pages 258–260. IEEE, 2022. (Cited at pages 159 e 169)

[182] Carlos Bermejo, Dimitris Chatzopoulos, and Pan Hui. Eyeshopper: Estimating shoppers' gaze using cctv cameras. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2765–2774, 2020. (Cited at page 159)

[183] Jonathan Fürst, Kaifei Chen, Hyung-Sin Kim, and Philippe Bonnet. Evaluating bluetooth low energy for iot. In *2018 IEEE Workshop on Benchmarking Cyber-Physical Networks and Systems (CPSBench)*, pages 1–6, 2018. (Cited at page 159)

[184] Jan Ližbetin and Jan Pečman. Possibilities of using bluetooth low energy beacon technology to locate objects internally: A case study. *Technologies*, 11(2), 2023. (Cited at page 159)

[185] Giridhar D. Mandyam, Mauro Scagnol, and Nicolas Graube. Secure onboarding and management of electronic shelf labels in retail. In *2023 15th International Conference on COMmunication Systems & NETworkS (COMSNETS)*, pages 96–101, 2023. (Cited at page 159)

[186] Vicente Cantón Paterna, Anna Calveras Augé, Josep Paradells Aspas, and María Alejandra Pérez Bullones. A bluetooth low energy indoor positioning system with channel diversity, weighted trilateration and kalman filtering. *Sensors*, 17(12), 2017. (Cited at page 164)

[187] Mehmet Cihan Sakman, Panagiotis Gkikopoulos, Francesco Martella, Massimo Villari, and Josef Spillner. Smart personalised shopping web prototype, April 2023. https://doi.org/10.5281/zenodo.7859411. (Cited at pages 169 e 170)

[188] Nuno Paulino, Luís M. Pessoa, André Branquinho, and Edgar Gonçalves. Design and experimental evaluation of a bluetooth 5.1 antenna array for angle-of-arrival estimation. In *2022 13th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, pages 625–630, 2022. (Cited at page 169)

[189] Panagiotis Gkikopoulos, Peter G. Kropf, Valerio Schiavoni, and Josef Spillner. AVOC: history-aware data fusion for reliable iot analytics. In Kaiwen Zhang, Abdelouahed Gherbi, and Paolo Bellavista, editors, *Proceedings of the 23rd International Middleware Conference: Industrial Track, Middleware 2022, Quebec, Quebec City, Canada, November 7-11, 2022*, pages 1–7. ACM, 2022. (Cited at page 171)

[190] Geoflyer 3d maps. `https://geoflyer3dmaps.com/`. Accessed: 2021-05-30. (Cited at page 172)

[191] Google maps. `https://maps.google.it`. Accessed: 2021-05-30. (Cited at page 172)

[192] Jing Du, Zhengbo Zou, Yangming Shi, and Dong Zhao. Zero latency: Real-time synchronization of bim data in virtual reality for collaborative decision-making. *Automation in Construction*, 85:51–64, 2018. (Cited at page 172)

[193] Vito Getuli, Pietro Capone, Alessandro Bruttini, and Shabtai Isaac. Bim-based immersive virtual reality for construction workspace planning: A safety-oriented approach. *Automation in Construction*, 114:103160, 2020. (Cited at page 173)

[194] Slava Kisilevich, Milos Krstajic, Daniel Keim, Natalia Andrienko, and Gennady Andrienko. Event-based analysis of people's activities and behavior using Flickr and Panoramio geotagged photo collections. *Proceedings of the International Conference on Information Visualisation*, pages 289–296, 2010. (Cited at page 174)

[195] Takashi Nicholas Maeda, Mitsuo Yoshida, Fujio Toriumi, and Hirotada Ohashi. Extraction of tourist destinations and comparative analysis of preferences between foreign tourists and domestic tourists on the basis of geotagged social media data. *ISPRS International Journal of Geo-Information*, 7(3), 2018. (Cited at page 174)

[196] Taylor Shelton. Spatialities of data: mapping social media 'beyond the geotag'. *GeoJournal*, 82(4):721–734, 2017. (Cited at page 174)

[197] Luke Sloan and Jeffrey Morgan. Who tweets with their location? Understanding the relationship between demographic characteristics and the use of geoservices and geotagging on twitter. *PLoS ONE*, 10(11):1–15, 2015. (Cited at page 174)

[198] Yusuke Nakaji and Keiji Yanai. Visualization of real-world events with geotagged tweet photos. *Proceedings of the 2012 IEEE International Conference on Multimedia and Expo Workshops, ICMEW 2012*, pages 272–277, 2012. (Cited at page 174)

[199] Ruofei Du and Amitabh Varshney. Social Street View: Blending Immersive Street Views with Geo-Tagged Social Media. pages 77–85, 2016. (Cited at page 174)

[200] Ruofei Du and David Li. Geollery : A Mixed Reality Social Media Platform Geollery : A Mixed Reality Social Media Platform. (April), 2019. (Cited at page 174)

[201] K. jr and B. Michalík. Laser scanning for bim and results visualization using vr. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-5/W2:49–52, 09 2019. (Cited at page 174)

[202] Berta Carrión-Ruiz, Silvia Blanco-Pons, M. Duong, J. Chartrand, M. Li, Kristine Prochnau, Stephen Fai, and José Lerma. Augmented experience to disseminate cultural heritage: House of commons windows, parliament hill national historic site (canada). *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W9:243–247, 01 2019. (Cited at page 175)

[203] Chongzheng Zhao and Lin Lv. Research on the feasibility of solving problems in construction of sponge city based on gis, vr and bim combined technology. *IOP Conference Series: Earth and Environmental Science*, 170:022068, 07 2018. (Cited at page 175)

[204] Jong-Won Lee, Deuk-Woo Kim, Seung-Eon Lee, and Jae-Weon Jeong. Development of a building occupant survey system with 3d spatial information. *Sustainability*, 12(23), 2020. (Cited at page 175)

[205] Google plus codes. `https://maps.google.com/pluscodes/`. Accessed: 2021-05-30. (Cited at page 177)

[206] Paulo Flores. *Global and Local Coordinates, In: Concepts and Formulations for Spatial Multibody Dynamics. Springer*, volume 168. 03 2015. (Cited at page 178)

[207] Europe 2020 - for a healthier EU. https://ec.europa.eu/health/europe_2020_en [accessed on 01/06/2019]. (Cited at pages 185 e 191)

[208] Telemedcare Clinical Monitoring Unit (CMU), Available online: https://www.telemedcare.com/ (accessed on 20 March 2020). (Cited at page 188)

[209] Enverse Continous Glucose Monitoring, Available online: https://www.eversensediabetes.com/ (accessed on 20 March 2020). (Cited at page 188)

[210] Med-care, Available online: http://www.t4all.it/portfolio-articoli/med-care/ (accessed on 20 March 2020). (Cited at page 188)

[211] HomoScreen, Available online: https://www.pixcell-medical.com/products/hemoscreen (accessed on 20 March 2020). (Cited at page 188)

[212] Samsung Labgeo PT10S, Available online: https://samsunghealthcare.com/en/products (accessed on 20 March 2020). (Cited at page 188)

[213] Md Kamrul Hasan, Md Hasanul Aziz, Md Ishrak Islam Zarif, Mahmudul Hasan, MMA Hashem, Shion Guha, Richard R Love, and Sheikh Ahamed. Noninvasive hemoglobin level prediction in a mobile phone environment: State of the art review and recommendations. *JMIR Mhealth Uhealth*, 9(4):e16806, 2021. (Cited at page 189)

[214] E.J. Wang, J. Zhu, W. Li, R. Rana, and S. Patel. Hemaapp ir: Noninvasive hemoglobin measurement using unmodified smartphone cameras and built-in leds. pages 305–308, 2017. (Cited at page 189)

[215] Non-invasive measurement of blood water content using infrared light. (Cited at page 189)

[216] Anika Tabassum Priyoti, Salman Jubair Jim, Sushmit Hossain, Shafkat Mahmud, Sujana Salvin, and Arnab Bhattacharjee. Non-invasive blood glucose measurement using near infra-red spectroscopy. In *2019 IEEE R10 Humanitarian Technology Conference (R10-HTC)(47129)*, pages 1–4, 2019. (Cited at page 189)

[217] Betty Elisabeth Manurung, Hugi Reyhandani Munggaran, Galih Fajar Ramadhan, and Allya Paramita Koesoema. Non-invasive blood glucose monitoring using near-infrared spectroscopy based on internet of things using machine learning. In *2019 IEEE R10 Humanitarian Technology Conference (R10-HTC)(47129)*, pages 5–11, 2019. (Cited at page 189)

[218] MD. Rezwanul Haque, S. M. Taslim Uddin Raju, MD. Asaf-Uddowla Golap, and M. M. A. Hashem. Corrections to "a novel technique for non-invasive measurement of human blood component levels from fingertip video using dnn based models". *IEEE Access*, 9:84178–84179, 2021. (Cited at page 189)

[219] Svenja Meyhöfer, Britta Wilms, Flavia Ihling, Anne Windjäger, Hannes Kalscheuer, Andrej Augustinov, Vera Herrmann, and Hendrik Lehnert. Evaluation of a near-infrared light ultrasound system as a non-invasive blood glucose monitoring device. *Diabetes, Obesity & Metabolism*, 22(4):694–698, 2020. (Cited at page 189)

[220] René van der Bel, Bart C. Sliggers, Marc J. van Houwelingen, Johannes J. van Lieshout, John R. Halliwill, Robert A. van Hulst, and C. T. Paul Krediet. A modified device for continuous non-invasive blood pressure measurements in humans under hyperbaric and/or oxygen-enriched conditions. *Diving and Hyperbaric Medicine*, 46(1):38–42, 2016. (Cited at page 189)

[221] Andrea Tiloca. *A machine learning approach for non-invasive blood pressure estimation*. laurea, Politecnico di Torino, March 2020. (Cited at page 190)

[222] Marinus Huber, Kosmas V. Kepesidis, Liudmila Voronina, Maša Božić, Michael Trubetskov, Nadia Harbeck, Ferenc Krausz, and Mihaela Žigman. Stability of person-specific blood-based infrared molecular fingerprints opens up prospects for health monitoring. *Nature Communications*, 12(1):1511, 2021. (Cited at page 190)

[223] FIWARE DATA MODELS. https://fiware-datamodels.readthedocs.io/en/latest/index.html. (Cited at pages 194 e 211)

[224] Alina Buzachis, Maria Fazio, Antonio Celesti, and Massimo Villari. Osmotic flow deployment leveraging faas capabilities. In Raffaele Montella, Angelo Ciaramella, Giancarlo Fortino, Antonio Guerrieri, and Antonio Liotta, editors, *Internet and Distributed Computing Systems*, pages 391–401, Cham, 2019. Springer International Publishing. (Cited at page 195)

[225] L. U. Khan, I. Yaqoob, N. H. Tran, S. M. A. Kazmi, T. N. Dang, and C. S. Hong. Edge computing enabled smart cities: A comprehensive survey. *IEEE Internet of Things Journal*, pages 1–1, 2020. (Cited at page 195)

[226] X. Xu, Q. Huang, X. Yin, M. Abbasi, M. R. Khosravi, and L. Qi. Intelligent offloading for collaborative smart city services in edge computing. *IEEE Internet of Things Journal*, pages 1–1, 2020. (Cited at page 195)

[227] N. A. M. Alduais, J. Abdullah, and A. Jamil. Rdcm: An efficient real-time data collection model for iot/wsn edge with multivariate sensors. *IEEE Access*, 7:89063–89082, 2019. (Cited at page 195)

[228] T. Rahman, X. Yao, G. Tao, H. Ning, and Z. Zhou. Efficient edge nodes reconfiguration and selection for the internet of things. *IEEE Sensors Journal*, 19(12):4672–4679, 2019. (Cited at page 195)

[229] I. Filip, A. V. Postoaca, R. Stochitoiu, D. Neatu, C. Negru, and F. Pop. Data capsule: Representation of heterogeneous data in cloud-edge computing. *IEEE Access*, 7:49558–49567, 2019. (Cited at page 195)

[230] Yousuke Watanabe, Kenya Sato, and Hiroaki Takada. Dynamicmap 2.0: A traffic data management platform leveraging clouds, edges and embedded systems. *International Journal of Intelligent Transportation Systems Research*, 18, 11 2018. (Cited at page 196)

[231] B. D. Cruz, J. Cheng, Z. Song, and E. Tilevich. Understanding the potential of edge-based participatory sensing: an experimental study. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pages 1–5, 2020. (Cited at page 196)

[232] A. J. Jara, Y. Bocchi, D. Fernandez, G. Molina, and A. Gomez. An analysis of context-aware data models for smart cities: Towards fiware and etsi cim emerging data model. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-4/W3:43–50, 2017. (Cited at page 196)

[233] Nathan Ota and William Kramer. Tinyml: Meta-data for wireless networks. 01 2021. (Cited at page 196)

[234] Read the Docs. FIWARE Data Models: NGSI-LD Device. Last access on February 1st, 2021. (Cited at page 197)

[235] Ahmed Abid, Jieun Lee, Franck Le Gall, and JaeSeung Song. Toward mapping an ngsi-ld context model on rdf graph approaches: A comparison study. *Sensors*, 22(13), 2022. (Cited at page 212)

[236] Darko Androvcec. Using json-ld to compose different iot and cloud services. 2018. (Cited at page 213)

[237] Yue Zhang and Christopher Stewart. Poster: Configuration management for internet services at the edge: A data-driven approach. In *2020 IEEE/ACM Symposium on Edge Computing (SEC)*, pages 155–157, 2020. (Cited at page 213)

[238] Hani Sami and Azzam Mourad. Dynamic on-demand fog formation offering on-the-fly iot service deployment. *IEEE Transactions on Network and Service Management*, 17(2):1026–1039, 2020. (Cited at page 213)

[239] M V Kumudavalli and G Venkatesh. Cloud computing based monolithic to containerization using elastic container service for phylogenetic analysis. In *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, pages 1540–1543, 2021. (Cited at page 213)

[240] Antonio Celesti, Davide Mulfari, Antonino Galletta, Maria Fazio, Lorenzo Carnevale, and Massimo Villari. A study on container virtualization for guarantee quality of service in cloud-of-things. *Future Gener. Comput. Syst.*, 99(C):356–364, oct 2019. (Cited at page 213)

[241] Massimo Villari, Maria Fazio, Schahram Dustdar, Omer Rana, and Rajiv Ranjan. Osmotic computing: A new paradigm for edge/cloud integration. *IEEE Cloud Computing*, 3(6):76–83, 2016. (Cited at page 213)

[242] Omogbai Oleghe. Container placement and migration in edge computing: Concept and scheduling models. *IEEE Access*, 9:68028–68043, 2021. (Cited at page 213)

[243] python-ngsild-client. https://github.com/Orange-OpenSource/python-ngsild-client. Last access: 03/12/2022. (Cited at page 219)

[244] Dejan Mijić and Ervin Varga. Unified iot platform architecture platforms as major iot building blocks. In *2018 International Conference on Computing and Network Communications (CoCoNet)*, pages 6–13. IEEE, 2018. (Cited at page 224)

[245] Antonio Cilfone, Luca Davoli, Laura Belli, and Gianluigi Ferrari. Wireless mesh networking: An iot-oriented perspective survey on relevant technologies. *Future Internet*, 11(4):99, Apr 2019. (Cited at page 224)

[246] S. Ali, M. Pandey, and N. Tyagi. Wireless-fog mesh: A framework for in-network computing of microservices in semipermanent smart environments. *International Journal of Network Management*, 30(6), 2020. (Cited at page 224)

[247] Xiaofan Jiang, Heng Zhang, Edgardo Alberto Barsallo Yi, Nithin Raghunathan, Charilaos Mousoulis, Somali Chaterji, Dimitrios Peroulis, Ali Shakouri, and Saurabh Bagchi. Hybrid low-power wide-area mesh network for iot applications. *IEEE Internet of Things Journal*, 2020. (Cited at page 224)

[248] Celia Garrido-Hidalgo, Diego Hortelano, Luis Roda-Sanchez, Teresa Olivares, M Carmen Ruiz, and Vicente Lopez. Iot heterogeneous mesh network deployment for human-in-the-loop challenges towards a social and sustainable industry 4.0. *IEEE Access*, 6:28417–28437, 2018. (Cited at page 224)

[249] Archit Gajjar, Xiaokun Yang, Hakduran Koc, Ishaq Unwala, Lei Wu, and Jiang Lu. Mesh-iot based system for large-scale environment. In *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 1019–1023. IEEE, 2018. (Cited at page 224)

[250] Emanuele Di Pascale, Irene Macaluso, Avishek Nag, Mark Kelly, and Linda Doyle. The network as a computer: A framework for distributed computing over iot mesh networks. *IEEE Internet of Things Journal*, 5(3):2107–2119, 2018. (Cited at page 224)

[251] Mennan Selimi, Adisorn Lertsinsrubtavee, Arjuna Sathiaseelan, Llorenç Cerdà-Alabern, and Leandro Navarro. Picasso: Enabling information-centric multi-tenancy at the edge of community mesh networks. *Computer Networks*, 164:106897, 2019. (Cited at page 225)

[252] Xiliu He and Fang Deng. Research on architecture of internet of things platform based on service mesh. In *2020 12th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, pages 755–759. IEEE, 2020. (Cited at page 225)

[253] Marino Linaje and Enrique Carlos Mesías. A new wsn mesh protocol for more transparent iot devices. In *International Workshop on Gerontechnology*, pages 94–106. Springer, 2018. (Cited at page 225)

[254] D. Huynh-Van, K. Tran-Quoc, and Q. LE-TRUNG. An empirical study on approaches of internet of things reconfiguration. In *2018 IEEE Seventh International Conference on Communications and Electronics (ICCE)*, pages 57–62, 2018. (Cited at page 225)

[255] Glusterfs official documentation. `https://docs.gluster.org`. Last access: December 29, 2020. (Cited at page 227)

[256] Yang Liu, Anbu Huang, Yun Luo, He Huang, Youzhi Liu, Yuanyuan Chen, Lican Feng, Tianjian Chen, Han Yu, and Qiang Yang. Fedvision: An online visual object detection platform powered by federated learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(08):13172–13179, Apr. 2020. (Cited at page 235)

[257] M. Mirmehdi, P. L. Palmer, and J. Kittler. Towards optimal zoom for automatic target recognition. Technical report, GBR, 1997. (Cited at page 236)

[258] Haoyu Duan, Yumin Zhang, and Wei Sheng. Image digital zoom based single target apriltag recognition algorithm in large scale changes on the distance. In *2019 1st International Conference on Industrial Artificial Intelligence (IAI)*, pages 1–6, 2019. (Cited at page 236)

[259] Mingfei Gao, Ruichi Yu, Ang Li, Vlad I. Morariu, and Larry S. Davis. Dynamic zoom-in network for fast object detection in large images, 2017. (Cited at page 236)

[260] Andreas Kolb, Erhardt Barth, Reinhard Koch, and Rasmus Larsen. Time-of-flight cameras in computer graphics. *Computer Graphics Forum*, 29:141 – 159, 03 2010. (Cited at page 236)

[261] Sicari, C. and Galletta, A. and Celesti, A. and Fazio, M. and Villari, M. An Osmotic Computing Enabled Domain Naming System (OCE-DNS) for distributed service relocation between cloud and edge. *Computers and Electrical Engineering*, 96, 2021. (Cited at pages 236 e 246)

[262] Wei Li, Xiang Sheng Feng, Kaiwen Zha, Shuning Li, and Hua Sheng Zhu. 1757(1):012003, jan 2021. (Cited at page 236)

[263] Yuwu Wang, Guobing Sun, and Shengwei Guo. Target detection method for low-resolution remote sensing image based on esrgan and redet. *Photonics*, 8(10), 2021. (Cited at page 237)

[264] Antonino Galletta, Armando Ruggeri, Maria Fazio, Gianluca Dini, and Massimo Villari. Mesmart-pro: Advanced processing at the edge for smart urban monitoring and reconfigurable services. *Journal of Sensor and Actuator Networks*, 9(4), 2020. (Cited at page 237)

[265] A. Mehrabi, M. Siekkinen, and A. Ylä-Jaaski. Qoe-traffic optimization through collaborative edge caching in adaptive mobile video streaming. *IEEE Access*, 6:52261–52276, 2018. (Cited at page 237)

[266] S. S. Kafılo˘glu, G. Gür, and F. Alagöz. Cooperative caching and video characteristics in d2d edge networks. *IEEE Communications Letters*, pages 1–1, 2020. (Cited at page 237)

[267] D. Wang, Y. Peng, X. Ma, W. Ding, H. Jiang, F. Chen, and J. Liu. Adaptive wireless video streaming based on edge computing: Opportunities and approaches. *IEEE Transactions on Services Computing*, 12(5):685–697, 2019. (Cited at page 237)

[268] Y. Lamdan and H.J. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *[1988 Proceedings] Second International Conference on Computer Vision*, pages 238–249, 1988. (Cited at page 240)

[269] Hwang-Soo Kim, R. Jain, and R. Volz. Object recognition using multiple views. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 28–33, 1985. (Cited at pages 240 e 242)

[270] Federated Learning - Google, url=`https://ai.googleblog.com/2017/04/federated-learning-collaborative.html`, note = Last access Sep 2022. (Cited at page 249)

[271] H Brendan McMahan Eider Moore Daniel Ramage Seth Hampson Blaise AgüeraAg and Agüera Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. 2017. (Cited at page 249)

[272] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. *2015 53rd Annual Allerton Conference on Communication, Control, and Computing, Allerton 2015*, pages 909–910, 4 2016. (Cited at page 249)

[273] Wei Liu, Li Chen, and Wenyi Zhang. Decentralized federated learning: Balancing communication and computing costs. *IEEE Transactions on Signal and Information Processing over Networks*, 8:131–143, 2022. (Cited at page 249)

[274] Ruchi Gupta and Tanweer Alam. Survey on Federated-Learning Approaches in Distributed Environment. *Wireless Personal Communications*, 2022. (Cited at page 249)

[275] H. Zhang, J. Bosch, and H.H. Olsson. Federated learning systems: Architecture alternatives. In *Proceedings - Asia-Pacific Software Engineering Conference, APSEC*, volume 2020-December, pages 385–394, 2020. (Cited at page 249)

[276] Collin Meese, Hang Chen, Syed Ali Asif, Wanxin Li, Chien-Chung Shen, and Mark Nejad. Bfrt: Blockchained federated learning for real-time traffic flow prediction. In *2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, pages 317–326, 2022. (Cited at page 249)

[277] Hassan Saadat, Abdulla Aboumadi, Amr Mohamed, Aiman Erbad, and Mohsen Guizani. Hierarchical federated learning for collaborative ids in iot applications. *2021 10th Mediterranean Conference on Embedded Computing, MECO 2021*, 6 2021. (Cited at page 249)

[278] Shen Xin, Li Zhuo, and Chen Xin. Online node cooperation strategy design for hierarchical federated learning. *INFOCOM WKSHPS 2022 - IEEE Conference on Computer Communications Workshops*, 2022. (Cited at page 249)

[279] Binxuan Hu, Yujia Gao, Liang Liu, and Huadong Ma. *Federated Region-Learning: An Edge Computing Based Framework for Urban Environment Sensing; Federated Region-Learning: An Edge Computing Based Framework for Urban Environment Sensing*. 2018. (Cited at page 249)

[280] Yujia Gao, Liang Liu, Binxuan Hu, Tianzi Lei, and Huadong Ma. Federated region-learning for environment sensing in edge computing system. *IEEE Transactions on Network Science and Engineering*, 7:2192–2204, 10 2020. (Cited at page 249)

[281] Abhijit Guha Roy, Shayan Siddiqui, Sebastian Pölsterl, Nassir Navab, and Christian Wachinger. Braintorrent: A peer-to-peer environment for decentralized federated learning. 5 2019. (Cited at page 250)

[282] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. Federated learning on non-iid data: A survey. *Neurocomputing*, 465:371–390, 11 2021. (Cited at page 250)

[283] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. (Cited at pages 250 e 253)

[284] Hyungbin Kim, Yongho Kim, and Hyunhee Park. Reducing model cost based on the weights of each layer for federated learning clustering. *International Conference on Ubiquitous and Future Networks, ICUFN*, 2021-August:405–408, 8 2021. (Cited at page 250)

[285] Cheng Chen, Ziyi Chen, Yi Zhou, and Bhavya Kailkhura. Fedcluster: Boosting the convergence of federated learning via cluster-cycling. *Proceedings - 2020 IEEE International Conference on Big Data, Big Data 2020*, pages 5017–5026, 12 2020. (Cited at page 250)

[286] Alessio Catalfamo, Antonio Celesti, Maria Fazio, Giovanni Randazzo, and Massimo Villari. A platform for federated learning on the edge: a video analysis use case. In *2022 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–7, 2022. (Cited at page 250)

[287] Armando Ruggeri, Antonio Celesti, Maria Fazio, and Massimo Villari. An innovative blockchain-based orchestrator for osmotic computing. *Journal of Grid Computing*, 20(1):1–17, 2022. (Cited at page 250)

[288] A. Celesti, D. Mulfari, A. Galletta, M. Fazio, L. Carnevale, and M. Villari. A study on container virtualization for guarantee quality of service in Cloud-of-Things. *Future Generation Computer Systems*, 99, 2019. (Cited at page 251)

[289] Kubernetes Lightweight by Rancher, howpublished = `https://k3s.io/`, note = Last access Oct 2022. (Cited at page 254)

[290] GlusterFS network filesystem, howpublished = `https://www.gluster.org/`, note = Last access Oct 2022. (Cited at page 254)

[291] Miltiadis D Lytras and Anna Visvizi. Who uses smart city services and what to make of it: Toward interdisciplinary smart cities research. *Sustainability*, 10(6):1998, 2018. (Cited at page 261)

[292] Svein Ølnes, Jolien Ubacht, and Marijn Janssen. Blockchain in government: Benefits and implications of distributed ledger technology for information sharing. *Government Information Quarterly*, 34(3):355–364, sep 2017. (Cited at page 261)

[293] A. Ruggeri, A. Celesti, M. Fazio, A. Galletta, and M. Villari. Bcb-x3dh: A blockchain based improved version of the extended triple diffie-hellman protocol. In *Proceedings - 2020 2nd IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications, TPS-ISA 2020*, pages 73–78, 2020. (Cited at page 261)

[294] Claudia Antal, Tudor Cioara, Ionut Anghel, Marcel Antal, and Ioan Salomie. Distributed ledger technology review and decentralized applications development guidelines. *Future Internet*, 13(3):62, 2021. (Cited at page 261)

[295] Christian Esposito, Massimo Ficco, and Brij Bhooshan Gupta. Blockchain-based authentication and authorization for smart city applications. *Information Processing and Management*, 58(2), mar 2021. (Cited at page 261)

[296] Victor Sucasas, Georgios Mantas, Ayman Radwan, and Jonathan Rodriguez. An OAuth2-based protocol with strong user privacy preservation for smart city mobile e-Health apps. In *2016 IEEE International Conference on Communications, ICC 2016*. Institute of Electrical and Electronics Engineers Inc., jul 2016. (Cited at page 262)

[297] Subramani Jegadeesan, Maria Azees, Priyan Malarvizhi Kumar, Gunasekaran Manogaran, Naveen Chilamkurti, R Varatharajan, and Ching-Hsien Hsu. An efficient anonymous mutual authentication technique for providing secure communication in mobile cloud computing for smart city applications. 2019. (Cited at page 262)

[298] Konstantinos Gerakos, Michael Maliappis, Constantina Costopoulou, and Maria Ntaliani. Electronic authentication for university transactions using eIDAS. In *Communications in Computer and Information Science*, volume 792, pages 187–195. Springer Verlag, 2017. (Cited at page 262)

[299] Aleksandr Ometov, Vitaly Petrov, Sergey Bezzateev, Sergey Andreev, Yevgeni Koucheryavy, and Mario Gerla. Challenges of multi-factor authentication for securing advanced iot applications. *IEEE Network*, 33(2):82–88, 2019. (Cited at page 262)

[300] Badr El Khalyly, Abdessamad Belangour, Mouad Banane, and Allae Erraissi. A comparative study of microservices-based iot platforms. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 11(7):389–398, 2020. (Cited at page 263)

[301] Nicola Dragoni, Saverio Giallorenzo, Alberto Lluch Lafuente, Manuel Mazzara, Fabrizio Montesi, Ruslan Mustafin, and Larisa Safina. Microservices: Yesterday, today, and tomorrow. *Present and Ulterior Software Engineering*, pages 195–216, 2017. (Cited at page 264)

[302] Alessio Catalfamo, Armando Ruggeri, Antonio Celesti, Maria Fazio, and Massimo Villari. A microservices and blockchain based one time password (MBB-OTP) protocol for security enhanced authentication. In *2021 IEEE Symposium on Computers and Communications (ISCC) (IEEE ISCC 2021)*, September 2021. (Cited at page 266)

[303] LXC, Available online: `https://linuxcontainers.org/` (accessed on 20 May 2021). (Cited at page 266)

[304] Infura, Available online: `https://infura.io/` (accessed on 20 May 2021). (Cited at page 269)

[305] European Parliaments. EU initiatives and funding to support sustainable urban mobility. 2020. Online; accessed 13 December 2021. (Cited at page 270)

[306] URBANITE. Decision making in the urban transformation field using disruptive technologies and a participatory approach. 2020. (Cited at page 271)

[307] Shincy Richu Jacob, Bibin Varghese, and Rangit Varghese. Cycling Management Using IoT – Keeping Track of Fitness Regime And Fall Detection. 2017. (Cited at page 271)

[308] Wan Norsyafizan W. Muhamad, Sayyidul Ainulfadhily bin Razali, Norfishah Ab Wahab, Meor Mohd Azreen, Suzi Seroja Sarnin, and Nani Fadzlina Naim. Smart bike monitoring system for cyclist via internet of things (iot). In *2020 IEEE 5th International Symposium on Telecommunication Technologies (ISTT)*, pages 168–173, 2020. (Cited at page 272)

[309] George Catargiu, Eva-H. Dulf, and Liviu C. Miclea. Connected bike-smart iot-based cycling training solution. *Sensors*, 20(5), 2020. (Cited at page 272)

[310] Esteban Municio, Glenn Daneels, Mathias De Brouwer, Femke Ongenae, Filip De Turck, Bart Braem, Jeroen Famaey, and Steven Latré. Continuous athlete monitoring in challenging cycling environments using iot technologies. *IEEE Internet of Things Journal*, 6(6):10875–10887, 2019. (Cited at page 272)

[311] Kyuhyun Lee and Ipek N. Sener. Emerging data for pedestrian and bicycle monitoring: Sources and applications. *Transportation Research Interdisciplinary Perspectives*, 4:100095, 2020. (Cited at page 272)

[312] Fabrizio Dabbene, Paolo Gay, and Cristina Tortia. Traceability issues in food supply chain management: A review. *Biosystems engineering*, 120:65–80, 2014. (Cited at page 287)

[313] Shanna Appelhanz, Victoria-Sophie Osburg, Waldemar Toporowski, and Matthias Schumann. Traceability system for capturing, processing and providing consumer-relevant information about wood products: system solution and its economic feasibility. *Journal of Cleaner Production*, 110:132–148, 2016. (Cited at page 287)

[314] Tarun Kumar Agrawal, Vijay Kumar, Rudrajeet Pal, Lichuan Wang, and Yan Chen. Blockchain-based framework for supply chain traceability: A case example of textile and clothing industry. *Computers Industrial Engineering*, 154:107130, 2021. (Cited at page 287)

[315] Kentaroh Toyoda, P Takis Mathiopoulos, Iwao Sasase, and Tomoaki Ohtsuki. A novel blockchain-based product ownership management system (poms) for anti-counterfeits in the post supply chain. *IEEE access*, 5:17465–17477, 2017. (Cited at page 287)

[316] Yi Lu, Peng Li, and He Xu. A food anti-counterfeiting traceability system based on blockchain and internet of things. *Procedia Computer Science*, 199:629–636, 2022. The 8th International Conference on Information Technology and Quantitative Management (ITQM 2020 2021): Developing Global Digital Economy after COVID-19. (Cited at page 287)

[317] INDICAM. Indicam. Online; accessed 29 May 2022. (Cited at page 287)

[318] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. May 2009. (Cited at page 287)

[319] Armando Ruggeri, Maria Fazio, Antonino Galletta, Antonio Celesti, and Massimo Villari. A decision support system for therapy prescription in a hospital centre. In *2020 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–4, 2020. (Cited at page 288)

[320] Jay Mehta, Darsh Mehta, Jainam Jain, and Surekha Dholay. Asset tracking system using blockchain. In *2021 Asian Conference on Innovation in Technology (ASIANCON)*, pages 1–7, 2021. (Cited at page 288)

[321] Ziyuan Wang, Lin Yang, Qin Wang, Donghai Liu, Zhiyu Xu, and Shigang Liu. Artchain: Blockchain-enabled platform for art marketplace. In *2019 IEEE International Conference on Blockchain (Blockchain)*, pages 447–454, 2019. (Cited at page 288)

[322] Luxochain. Luxochain. Online; accessed 03 July 2022. (Cited at page 288)

[323] Jaime José Cueva-Sánchez, Aaron Jair Coyco-Ordemar, and Willy Ugarte. A blockchain-based technological solution to ensure data transparency of the wood supply chain. In *2020 IEEE ANDESCON*, pages 1–6, 2020. (Cited at page 288)

[324] Sandi Rahmadika, Bruno Joachim Kweka, Cho Nwe Zin Latt, and Kyung-Hyune Rhee. A preliminary approach of blockchain technology in supply chain system. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 156–160, 2018. (Cited at page 288)

[325] Gabin Heo, Dana Yang, Inshil Doh, and Kijoon Chae. Efficient and secure blockchain system for digital content trading. *IEEE Access*, 9:77438–77450, 2021. (Cited at page 288)

[326] Jack Davies and Yingli Wang. Physically unclonable functions (pufs): A new frontier in supply chain product and asset tracking. *IEEE Engineering Management Review*, 49(2):116–125, 2021. (Cited at page 289)

[327] Joshua A. BakerPhilipp Fuhrmann Tarek I. Saab, Bruce Kleinman. Tracking and verifying authenticity of an asset via a distributed ledger, 2019. US Patent. (Cited at page 289)

[328] fondapol. Blockchain and distributed trust. Online; accessed 02 May 2022. (Cited at page 289)

[329] Dejan Vujičić, Dijana Jagodić, and Siniša Ranđić. Blockchain technology, bitcoin, and ethereum: A brief overview. In *2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH)*, pages 1–6, 2018. (Cited at page 289)

[330] deloitte. The global power of luxury. Online; accessed 05 May 2022. (Cited at page 289)

[331] Ethereum. Ethereum documentation. Online; accessed 26 April 2022. (Cited at page 293)

[332] P©ter Heged±s. Towards analyzing the complexity landscape of solidity based ethereum smart contracts. *Technologies*, 7(1), 2019. (Cited at page 293)

[333] Rajitha Yasaweerasinghelage, Mark Staples, and Ingo Weber. Predicting latency of blockchain-based systems using architectural modelling and simulation. In *2017 IEEE International Conference on Software Architecture (ICSA)*, pages 253–256, 2017. (Cited at page 294)

[334] Cátia Santos-Pereira, Alexandre B. Augusto, Ricardo Cruz-Correia, and Manuel E. Correia. A secure rbac mobile agent access control model for healthcare institutions. In *Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems*, pages 349–354, 2013. (Cited at page 295)

[335] Alvaro Alonso, Alejandro Pozo Huertas, Johnny Choque, Gloria Bueno, Joaquin Salvachua, Luis Diez, Jorge Marin, and Pedro Alonso. An identity framework for providing access to fiware oauth 2.0-based services according to the eidas european regulation. *IEEE Access*, 7:88435 – 88449, 07 2019. (Cited at page 295)

[336] Vinayak Singla, Indra Kumar Malav, Jaspreet Kaur, and Sumit Kalra. Develop leave application using blockchain smart contract. In *2019 11th International Conference on Communication Systems Networks (COMSNETS)*, pages 547–549, 2019. (Cited at page 295)

[337] Shuai Wang, Liwei Ouyang, Yong Yuan, Xiaochun Ni, Xuan Han, and Fei-Yue Wang. Blockchain-enabled smart contracts: Architecture, applications, and future trends. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(11):2266–2277, 2019. (Cited at page 302)

[338] Fáber D. Giraldo, Barbosa Milton C., and Carlos E. Gamboa. Electronic voting using blockchain and smart contracts: Proof of concept. *IEEE Latin America Transactions*, 18(10):1743–1751, 2020. (Cited at page 302)

[339] Ilhaam A. Omar, Raja Jayaraman, Mazin S. Debe, Khaled Salah, Ibrar Yaqoob, and Mohammed Omar. Automating procurement contracts in the healthcare supply chain using blockchain smart contracts. *IEEE Access*, 9:37397–37409, 2021. (Cited at page 302)

[340] Akanksha Saini, Qingyi Zhu, Navneet Singh, Yong Xiang, Longxiang Gao, and Yushu Zhang. A smart-contract-based access control framework for cloud smart healthcare system. *IEEE Internet of Things Journal*, 8(7):5914–5925, 2021. (Cited at page 302)

[341] Abdullah Ayub Khan, Asif Ali Wagan, Asif Ali Laghari, Abdul Rehman Gilal, Izzatdin Abdul Aziz, and Bandeh Ali Talpur. Biomt: A state-of-the-art consortium serverless network architecture for healthcare system using blockchain smart contracts. *IEEE Access*, 10:78887–78898, 2022. (Cited at page 302)

[342] Anusha Vangala, Anil Kumar Sutrala, Ashok Kumar Das, and Minho Jo. Smart contract-based blockchain-envisioned authentication scheme for smart farming. *IEEE Internet of Things Journal*, 8(13):10792–10806, 2021. (Cited at page 302)

[343] Hari Pranav A, M. Latha, Ashwin. M. S, and R. Chinnaiyan. Blockchainas a service (baas) framework for government funded projects e-tendering process administration and quality assurance using smart contracts. In *2021 International Conference on Computer Communication and Informatics (ICCCI)*, pages 1–4, 2021. (Cited at page 302)

[344] Archana Sahai and Rajiv Pandey. Smart contract definition for land registry in blockchain. In *2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT)*, pages 230–235, 2020. (Cited at page 302)

[345] Fardin Ahmed Niloy, Md. Abu Nayeem, Md. Majedur Rahman, and Md. Nozib Ud Dowla. Blockchain-based peer-to-peer sustainable energy trading in microgrid using smart contracts. In *2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, pages 61–66, 2021. (Cited at page 302)

[346] Natthanan Chanthong, Thitiwat Ruangsakorn, and Sorayut Glomglome. Blockchain and smart contract payment for electric vehicle charging. In *2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 161–164, 2020. (Cited at page 303)

[347] Anni Karinsalo and Kimmo Halunen. Smart contracts for a mobility-as-a-service ecosystem. In *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pages 135–138, 2018. (Cited at page 303)

[348] JiaChen Hou, WenWen Ding, Xiaolong Liang, FengHua Zhu, Yong Yuan, and FeiYue Wang. A study on decentralized autonomous organizations based intelligent transportation system enabled by blockchain and smart contract. In *2021 China Automation Congress (CAC)*, pages 967–971, 2021. (Cited at page 303)

La borsa di dottorato è stata cofinanziata con risorse del
Programma Operativo Nazionale Ricerca e Innovazione 2014-2020 (CCI 2014IT16M2OP005),
Fondo Sociale Europeo, Azione I.1 "Dottorati Innovativi con caratterizzazione Industriale"

UNIONE EUROPEA
Fondo Sociale Europeo

Ministero dell'Università
e della Ricerca

PON
RICERCA
E INNOVAZIONE
2014 - 2020