**Università degli Studi di Salerno**

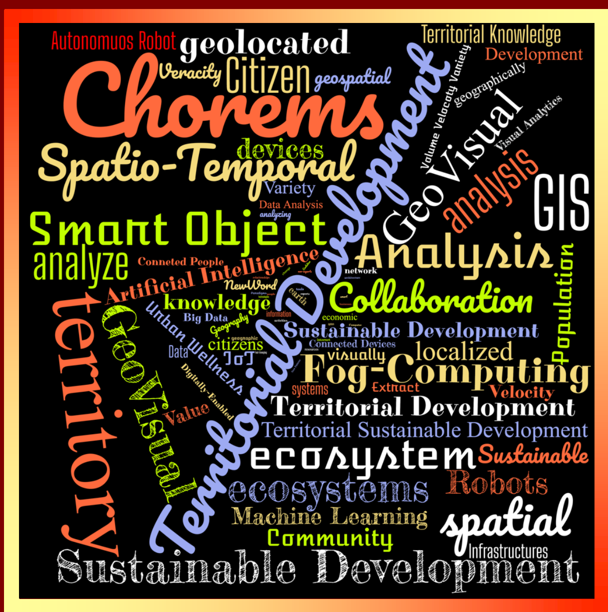
**Dipartimento di Informatica**

**Dottorato di Ricerca in Informatica – XXXIV Ciclo**

Tesi di Dottorato/Ph.D. Thesis

# GeoVisual Analytics methods and techniques for territorial sustainable development

**Pietro Battistoni**




Supervisor: **Prof. Monica Sebillo**

Ph.D. Program Director: **Prof. Andrea De Lucia**

**AA 2020/2021**

**Curriculum Internet of Things and Smart Technologies**

University of Salerno

Department of Computer Science

Dottorato di Ricerca in Informatica
XXXIV Ciclo

Tesi di Dottorato / Ph.D. Thesis

# GeoVisual Analytics methods and techniques for territorial sustainable development

**Pietro BATTISTONI**

Supervisor: **Prof.ssa Monica SEBILLO**

PhD Program Director: **Prof. Andrea DE LUCIA**

A.A 2020/2021

# ABSTRACT

In order to reduce poverty and improve people's lives everywhere, territorial development must be sustainable. Data science and analytics can offer fundamental contributions to it. Moreover, to achieve the 17 goals in the 2030 Agenda for Sustainable development defined by United Nations, citizens' participation in decision processes is paramount. Indeed, it helps fight against corruption and facilitates the policymakers aligning their decision with the needs of citizens.

The research described in this thesis has two main objectives, namely to harness Big Data for sustainable spatial development and to promote citizen involvement in decision-making processes by offering them a new tool for geovisual analysis of spatial data.

The growing number of devices and people connected to the Internet are valuable sources of geographically localized information, not yet fully exploited. Although this data has all the characteristics of the Big Data for the Sustainable Development, as defined by United Nations, it requires innovative architecture and tools to be managed. Furthermore, citizens' involvement requires that these tools should be simple to use and understand, even during complex data analyses.

The dissertation describes the research that led to the layered framework that helps leverage Big Data for Sustainable Development, and the new visual tool that helps citizens participate in data analysis.

---

Per ridurre la povertà e migliorare la vita delle persone, ovunque esse vivano, lo sviluppo territoriale deve essere sostenibile. La scienza dei dati e l'analisi possono offrire contributi fondamentali in tal senso. Inoltre, per raggiungere i 17 obiettivi dell'Agenda 2030 per lo sviluppo sostenibile definiti dall'ONU, la partecipazione dei cittadini ai processi decisionali è fondamentale, infatti, aiuta a combattere la corruzione e facilita i responsabili politici ad allineare le loro decisioni con le esigenze dei cittadini.

La ricerca descritta in questa tesi ha due obiettivi principali: sfruttare i Big Data per uno sviluppo territoriale sostenibile e promuovere il coinvolgimento dei cittadini nei processi decisionali offrendo loro un nuovo strumento per l'analisi geovisiva dei dati territoriali.

Il numero crescente di dispositivi e persone connesse a Internet è una fonte preziosa di informazioni geograficamente localizzate, non ancora pienamente sfruttate. Sebbene questi dati abbiano tutte le caratteristiche dei Big Data per lo sviluppo sostenibile, secondo la definizione dell'United Nations, richiedono un'architettura e strumenti innovativi per essere gestiti. Inoltre, il coinvolgimento dei cittadini richiede che questi strumenti siano semplici da usare e da capire, anche durante le analisi di dati complessi.

La tesi descrive la ricerca che ha portato alla struttura stratificata che aiuta a sfruttare i Big Data per lo sviluppo sostenibile e il nuovo strumento visivo che aiuta i cittadini a partecipare all'analisi dei dati.

# CONTENTS

# LIST OF FIGURES

# ACRONYMS

ADSL Asymmetric Digital Subscriber Line

ADV BLE advertise

AI Artificial Intelligence

AmI Ambient Intelligence

ANS Alert Notification Services

AP Access Point

API Application Programming Interface

BDSD Big Data for Sustainable Development

BLE Bluetooth Light Energy

CRUD Create, Read, Update, Delete

CSS Cascading Style Sheets

DB Data Base

DBMS Database Management Systems

DD-ST Description and Dictionary data, and Scheduled Trajectories

EAQI European Air Quality Index

EEC European Economic Community

EI Edge-Intelligence

EO Earth Observation

ETL Extract, Transform, Load

FMIS Farm Management Information Systems

GAP Generic Access Profile

GPIO General Purpose Input/Output Ports

GPS Global Positioning System

GSM Global System for Mobile Communications

GUI Graphic User Interface

GVA GeoVisual Analytics

HCI Human-Computer Interaction

HPC High-Performance computing

HTTP Hypertext Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

I2C Inter-Integrated Circuit

ICT Communication Technologies

IDE Integrated Development Environment

INRAE French National Research Institute for Agriculture, Food and Environment

INS Indoor Navigation System

IoT Internet of Things

IP Internet Protocol

IPS Indoor Positioning System

IT Information Technology

ISTAT National Institute of Statistics

LAMP Linux, Apache, MySQL, and PHP

LAN Local Area Network

LiDAR Light Detection and Ranging
LL Living Lab
LLs Living Labs
M2M Machine-to-Machine
MCU Microcontroller Unit
MILP mixed integer linear program
ML Machine Learning
MobEC Mobile Edge Computing
MPI Mazziotta-Pareto Index
MQTT An MQ Telemetry Transport
NDVI Normalized Difference Vegetation Index
NMEA National Marine Electronics Association
OG Open Government
OLAP Online Analytical Processing
OSH Occupational Health and Safety
PAN Personal Area Network
PC Personal Computer
PPE Personal Protective Equipment
Pub/Sub Publish / Subscribe
QoS Quality of Service
RFI Remote Function Invocation
RFID Radio-Frequency Identification
RL Reinforcement Learning
RMI Remote Method Invocation
ROS Robot Operating System
RPC Remote Procedure Call
RSSI Received Signal Strength Indication
S3 Strategies for Smart Specialisation
SDGs Sustainable Development Goals
SDK Software Development Kit

SDMX Statistical Data and Metadata eXchange

SMS Short Message Service

SoC System on Chip

SPI Serial Peripheral Interface

SSE Server-Sent Events

SSID Service Set Identifier

SSL Secure Sockets Layer

STA Wi-Fi Station Interface

TI Territorial Intelligence

TLS Transport Layer Security

UHI Urban Heat Island

ULP Ultra-Low Power co-processor

UN United Nations

VPN Virtual Private Network

WAN Wide Area Network

Web-GIS Geographic Information System on Web

WLAN Wireless Local Area Network

WPA2 Wi-Fi Protected Access II

# INTRODUCTION

The growing number of devices and people connected to the Internet represents a valuable source of geographically localized information not yet fully exploited. According to the Cisco Annual Internet Report (2018–2023), the number of connected devices, by 2023, will be more than three times the world's population. [31]. As result, 29.3 billion networked devices will produce data almost continuously. This heterogeneous data, produced in large quantities and at high speeds, qualifies as Big Data.

As recognized by United Nations (UN) [90], the analysis of this data can produce vital information for human settlements and their sustainable development. Therefore, the decision-makers should leverage this information, especially for real-time awareness.

## ISSUES AND CHALLENGES

Dealing with Big Data means dealing with data that has characteristics of:

- Volume, i.e., the amount of data to manipulate is huge.
- Velocity, i.e.,the huge amount of data is due to the number of sources available that generate data at a high rate.
- Variety, i.e., the data can be unstructured, semi-structured or structured. However, even when it is structured, each source can have a very different structure.

Big Data for Sustainable Development (BDSD) is Big Data that can help achieve the Sustainable Development Goals (SDGs) and be relevant to the policy and planning of sustainable development of the territory, according to the authors of [104], it must be *Geographically and Temporally Trackable*, *Digitally Generated*, *Automatically Collected*, and *Continuously Analyzed*. It implies that:

- data should be spatiotemporal;

- systems must automatically manipulate data;
- systems must extract and store relevant information;
- systems must analyze data on human welfare and development in real-time.

Collecting Big Data is already a challenge, but making it available and usable for the public good is even more so.

Spatiotemporal data analysis requires sophisticated tools to deal with multiple dimensions, trends, and predictions, and these tools are usually hard to use by non-experts in Information Technology (IT).

Experts use tools and methods that vary by domain and experience, but urban or rural inhabitants are excluded when the tools are not entirely intuitive and easy to use. Thus, it is still promising to investigate more generic and easy-to-use tools , to promote the citizen involvement in the analytical process and allow them to interact with the data in a meaningful way.

Furthermore, analyzing data from multiple sources in real-time requires new strategies. This data produced at different rates must be synchronized to have temporal consistency. It requires innovative methods and technologies to decouple producers from consumers.

Citizens' involvement in local and regional governments is paramount to address the SDGs of the UN Agenda 2030 [125]. It helps fight against corruption and facilitates the policymakers aligning their decision with the needs of citizens. Furthermore, accessible information helps citizens understand the need for some hard-to-accept decisions. The recent COVID-pandemic period demonstrates it.

Involving an ample number of citizens collaborating with experts and decision-makers for sustainable development of their territory demands new tools usable by experts and not. To reach this audience these tools should be accessible and searchable through modern web technologies. They must be accurate enough in data analysis, but easy to understand and use, even by ordinary people. Achieving this will produce an effective collaboration among all stakeholders of territory development, making a step ahead in the goals of SDGs.

## RESEARCH GOALS

The research described in this thesis aims to offer support to policymakers by adding the important contribution that Big Data for sustainable development could make, as well as proposing tools that are easy to use even by non-experts to inform citizens and promote their involvement in decisions.

This Big Data are overabundant stimuli even for experts with remarkable reasoning skills, thus, computational analysis narrows the field of what needs to be considered visually by humans. Furthermore, combining it with human interaction, through manipulation of visual components of the system, activates human analytical reasoning that identifies and defines problems, extracts key information from the data, and develops workable solutions to the identified problems [60]. Thus, the goal of this research was to support decision makers with a new visual model that allowed human interaction, manipulating the visual components of the system in an intuitive and simple manner that even the non-domain-savvy citizen would not be discouraged by using it.

In order to pursue this objective, the work has been focused on two main research domains, namely the Internet of Things (IoT) and GeoVisual Analytics (GVA).

The study of IoT and its architecture had the goal of leveraging these data sources for the sustainable development of the territory, experimenting with popular and innovative technologies for batch and real-time data collection from distributed devices.

The GVA can be defined as "the science of analytical reasoning facilitated by interactive visual interfaces" [120], then the study of GVA aimed to provide innovative tools to make complex analytical tasks visually interactive and easy to use by both technicians and ordinary citizens.

The research questions defined can be summarized as follows:

- RQ1 - How to improve methods and techniques for collecting large volumes of spatiotemporal data when a large network of connected devices is involved?
- RQ2 - How to process this data when a real-time awareness is required?

- RQ3 - How to improve the geovisual analysis tools for BDSD?

## METHODOLOGY AND TECHNIQUES

Through deploying some use cases, the most popular IoT architectural solutions for collecting data from connected devices were analyzed, and the most promising ones were explored in depth. As for the geovisual domain, initially the research focused on user expectations by investigating modern proposals for visual analytical tools. Then, by examining the problems that such tools have kept open, new theoretical concepts were defined to transform some existing visual metaphors into interactive and usable tools for visual analysis of geographic information. Finally, a new framework was proposed to visually explore and analyze BDSD of territory. This framework intends to be used to implement architectures and tools to dig and analyze territorial information as support to decision-makers and an instrument to evaluate what-if scenarios.

## RELEVANCE OF RESEARCH

The research results are new and relevant to the leverage of BDSD. They determined technical advances defining visual tools to explore this kind of data and perform analysis, even in real-time, offering substantial value for real-time awareness of the territorial situation, especially in emergencies.

## STRUCTURE OF THE THESIS

This dissertation is divided into two parts, recalling the two phases of the research. The former describes work related to the IoT domain. The letter presents new tools for visualization and analysis of geographic information. Several chapters report on the activities carried out during the two phases; in particular, Part I presents some work published during the three-year doctoral program, in which the IoT architectures implemented and the conclusions drawn are presented. Part II presents some works published during the same period, describing the studies carried out on the domain of GVA, followed by the description of the new proposal and the goals achieved.

## PART I - THE AGE OF EDGE-COMPUTING

Introduction to PART I: The Edge Computing Context - It describes the context relevant to the IoT research work, in common to the rest of the dissertation.

Chapter 1: Experimenting with Fog-Computing Architecture - It describes the experimentation with an indoor navigation system and notification service as a use case of a Fog-Computing architecture.

Chapter 2: Handling trillions of events a day - It describes the research work made in collaboration with INRAE, the French National Research Institute for Agriculture, Food and Environment. The work consisted of experiments with a Fog-Computing architecture to collect and visualize spatiotemporal data produced by Autonomous Robots running at INRAE experimental farm.

Chapter 3: Computation offloading - It describes a different use of the Publisher / Subscribe protocol.

Chapter 4: Smart Objects as sensors network based on Fog-Computing - It presents the Fog-Computing paradigm in collecting sensor data and locations, even when they are mobile and wearable.

## PART II- GEOVISUAL ANALYSIS

Chapter 5: An open innovation platform to support local development policies. - It proposes web platforms to inform and involve all stakeholders during decision-making processes and the development of territorial solutions facilitating the citizens' participation.

Chapter 6: Interactive maps - They are an easy and effective way to visualize geographic data, and this chapter enhances them by experimenting with new visual means.

Chapter 7: Actionable Chorems - It proposes an evolution of *Chorems* on computer displays by moving them from a visual analytical tool to an actionable object on geographic maps.

Chapter 8: Conclusions. - This chapter summarizes the content of the thesis, highlighting the research contribution.

Chapter 9: Future Work. - It is an overview of planned future work.

Part I

# THE AGE OF EDGE-COMPUTING

In the recent past, due to the increasing number of Internet Protocol (IP) connected devices, a new network architecture paradigm has become popular in the literature. Namely, the Edge-Computing paradigm promises to mitigate some problems associated with unreliable and costly Internet connections [114]. The idea is that moving the data analysis closer to the data source reduces the latency caused by the connection between data sources and computing devices.

Edge-Computing refers to a network architecture that involves devices located at the Edge of the Network. These devices must be capable of offering computational power and connectivity services to other devices on the network. The Edge of the Network is the boundary between the end device and the network. This boundary often extends across the entire local network, as illustrated in Figure 1.

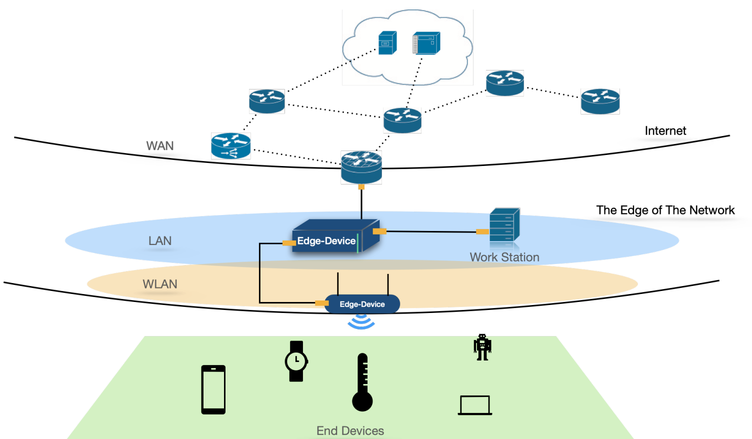



Figure 1: An Edge-Computing architecture example

As Figure 1 shows, it is possible to draw the connectivity between end-devices and Cloud resources in layers. Each intermediate layer connects the lower layer with the upper one and possibly performs intermediate processing before forwarding the data, thus distributing part of the processes on different nodes. For it, nodes must have resources capable of processing and connect.

This new architecture is possible because on market there is new hardware that offers adequate computing power, low energy consumption, and low operational costs [27].

Anyhow, Edge-Computing is not the alternative to Cloud architectures. Although edge devices have low costs and consume low energy, they have limited resources compared with Cloud resources. It is further true about storage resources. So, to store large amounts of data or use ultra-high computing power, Cloud resources are still the best option. Thus, in most cases, is preferred a mix of both architectures.

## FOG-COMPUTING

In 2010 at the CISCO/MBARI meeting, Flavio Bonomi presented his different vision of *Edge-Computing*. Ginny Nichols, an attendee at the meeting, suggested the term "Fog-Computing" to emphasize that "while clouds are high in the sky, fog is close to the ground, i.e., close to the users," as she explained to me.

The NIST Fog Computing Model [68] defines *Fog-Computing* as a layered model that facilitates the deployment of distributed applications and services, enables low latency, reliable operations, and removes the requirement for persistent cloud connectivity. Spanning between *End-Devices* and *Cloud-Resources* (Figure 2,) *Fog-Computing* components are *Fog-Nodes* that, in addition to offering traditional connection services, also offer computational services and, in some cases, storage.

There are fundamental differences between Edge-Computing and Fog-Computing.

Edge-Computing is the network layer that contains *End Devices* and their users. It provides local computing capabilities on sensors or other devices accessible from the network.

Fog-Computing runs applications in a multi-level architecture that decouples and connects hardware and software functions, allowing dynamic reconfiguration for different applications, and runs intelligent computing and communication services.

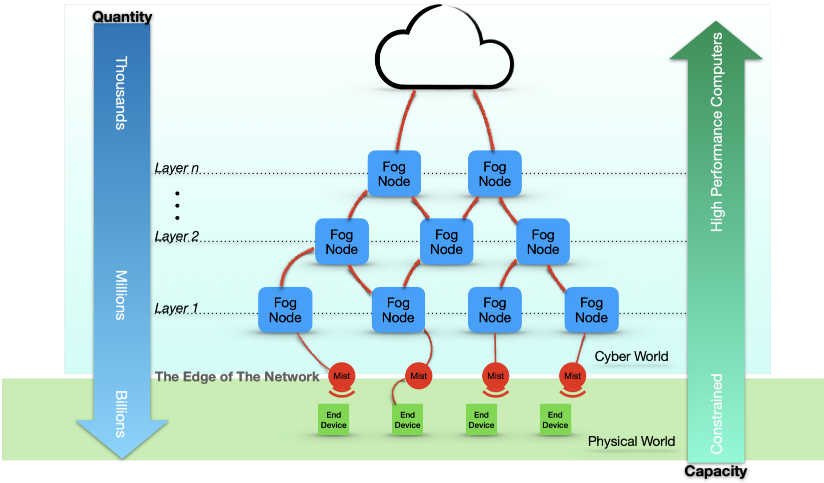



Figure 2: Fog Computing Architecture example

Edge-Computing executes specific applications in a fixed logical location, tends to be limited to a few peripheral devices, and provides a direct broadcast service.

Fog-Computing is hierarchical and addresses the storage, control, and acceleration of data processing in addition to computation and networking.

As shown in Figure 2, the number of connected *End-Devices* is literally billions. From the bottom up, the number of connected devices at each level decreases while their computing and storage capacity increases from constrained devices up to High-Performance computing (HPC).

A *Fog-Node* supplies end-devices with some form of data processing, computing resource, and communication services between network layers. In some cases, Fog-Nodes may also offer data storage. A Fog-Node operates in both centralized and decentralized ways. It can be, configured as a stand-alone, communicating with other nodes to deliver services, or federated to form clusters providing horizontal scalability over diffuse geolocations. *Fog-Nodes* when geographically distributed must be aware of both their geographical position and logical location within the context of their cluster.

Furthermore, [68] defines *Mist-Nodes* as *Fog-Nodes* with lower computational resources and more specialised. They are placed directly within the edge of the network fabric, closer to the *End-Devices* feeding the *Fog-nodes*. Often, they share the same locality with the smart *End-Devices* to which they give connectivity and computational services with lower latency.

The *Fog-Computing* can be thought of as a continuous data transformation into knowledge. From the bottom up at each layer, the data analysis decreases the data quantity and increases its information. Furthermore, this information can be accessed at any layer by other *Fog-Nodes*, thus reducing the path and latency between data producers and data consumers.

In [62] can be found other definitions and specifications produced by the OpenFog Consortium. It is very active in promoting the adoption of Fog-Computing.

In this dissertation, the Fog-Mist Computing paradigm is present in many of the following works.

# 1 EXPERIMENTING WITH FOG-COMPUTING ARCHITECTURE

## 1.1 SUMMARY

This chapter introduces the initial work done to study IoT architectures. It describes a solution implemented to enable an indoor navigation system on ships. The solution is designed to be used primarily in emergency situations that may require evacuation of the ship. The work provided an opportunity to design and improve the use of modern IoT architecture in constrained environments and in situations where system resilience is critical. It also experimented with asynchronous communication protocols capable of collecting data in real or near real-time.

## 1.2 INTRODUCTION

In [17], we propose experimentation with an indoor navigation system and notification service as a use case, of a Fog-Computing architecture. The purpose of the work was to implement a real solution, experiment with it, and learn in practice the advantages and problems that can be faced.

Figure 1.1 depicts the layered model, numbering the layers from bottom to up.

**Layer 1** consists of partially meshed hardware devices at the very edge of the network. Devices at Layer 1 connect the user's mobile devices with that in Layer 2, while a partially meshed network chains all devices in Layer 1. Due to the functionality and computational capacity, devices at Layer 1 are referred to here as *Mist-Nodes*.

One of the *Mist-Nodes* in Layer 1 is defined as *Root Mist-Node* and connects its Wi-Fi Station Interface (STA) to the Wi-Fi Access Point (AP) in Layer 2 (Figure 1.2).

The device in Layer 2 connects devices of Layer 1 with the Internet, reaching the Cloud. Due to the functionality and com-

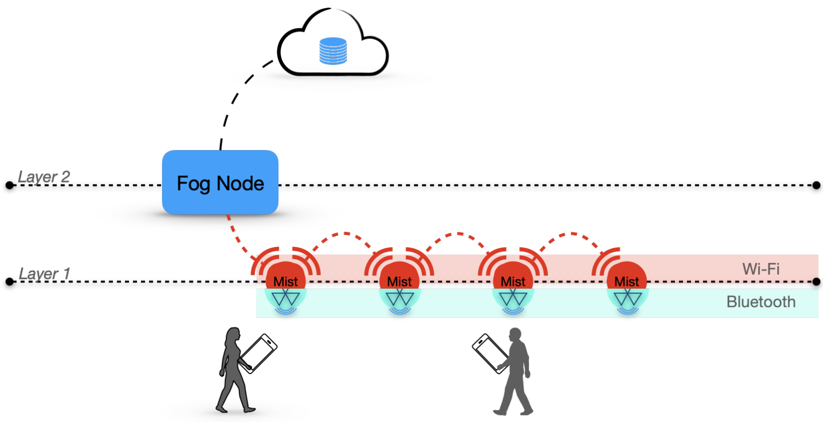



Figure 1.1: The Framework scheme for the indoor navigation system and notification service

putational capacity of its device, Layer 2 is referred to here as *Fog-Layer*

The STA interface of next Mist-Node in Layer 1 connects to the Soft AP interface of the root Mist-Node. The Soft AP interface of the remaining Mist-Nodes in Layer 1 accepts connections from STA interface of next mist node and from the below nearest user's mobile devices if any. Then, each Mist-Node can transmit its packets to previous Mist-Node and at the same time serves as a relay for the next node and the user-devices, as depicted in Figure 1.2

At Layer 1, ESP32-WROVER module-based devices are deployed. This module made by the Espressif Systems has the Wi-Fi and Bluetooth Light Energy (BLE) coexistence. The ESP32 is highly programmable and low cost (less than 4$ in a single piece) System on Chip (SoC). It integrates many interfaces in a single chip and consumes very low power. The "ESP32 Technical Reference Manual" [48] gives more details. Despite its two Xtensa® 32-bit LX6 Microprocessors and the Ultra-Low Power co-processor (ULP), it is a right candidate for the Mist-Nodes of the proposed model. The Espressif IoT Development Framework, based on ESP-IDF Software Development Kit (SDK) [118], offers a good set of Application Programming Interface (API)s to deploy

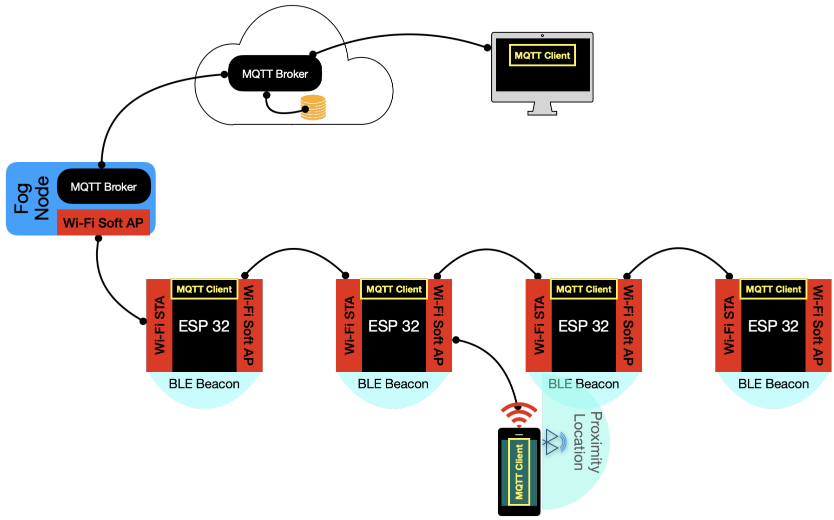



Figure 1.2: The Fog-Mist nodes scheme

on the SoC the firmware for the mesh network (ESP-MESH), and BLE functionalities. The partially meshed network of Mist-Nodes can have an extensible coverage area, as only one of Mist-Nodes is required to connect to the central AP. The Wi-Fi Network is also less susceptible to overloading, as a single AP no longer limits the number of end-user mobile devices allowed on the network.

The end-user mobile devices are defined as Leaf-Node since they cannot have any child nodes, which means no downstream connections. They can only transmit or receive their packets. In the proposed framework, the Leaf-Nodes join the network, connecting to the Mist-Node, which advertises the strongest Received Signal Strength Indication (RSSI) from its SoftAP interface. The Root Node is the Mist-Node designed as Root during configuration, or dynamically elected, based on the signal strength estimated between each node and the Fog-Node of Layer 2.

Layer 1 is a multi-hop network. Each Mist-Node device starts in an idle state, listening for Wi-Fi beacon frames. It generates a list of potential parents and forms an upstream with the strongest RSSI, moving out from the idle state. Building the mesh network can be autonomous and flexible, where Mist-Nodes can dynamically leave or join the network.

At Layer 2 the Raspberry Pi 3 Model B+ is deployed as a Fog-Node. This device has an ARMV8 64-bit SoC @ 1.4GHz, Wi-

Fi, Bluetooth, BLE, Gigabit Ethernet connectivity, and supports Micro SD cards to store operating system and data, for a cost of less than 50$. The officially supported operating system, the Raspbian, comes with plenty of software for programming and general purposes. All typical functionalities of a Wi-Fi Access Point are deployed on it.

### 1.3 A SCENARIO FOR INDOOR NAVIGATION

The described framework is used to design an Indoor Navigation System (INS), with a mobile end-user notification system, which allows users to send messages to each other, and with a centralized system. In order to realize the INS, it is mandatory to know the user's position in real-time. To reach this goal, an Indoor Positioning System (IPS) is proposed, where BLE beacons [39], transmitted from Layer 1 nodes, are used on a proximity-based strategy, allowing the end-user application to know its own position on a map and suggest the convenient path to reach the destination.

**An MQ Telemetry Transport (MQTT)** protocol solution is assumed, in order to realize the notification system. It is a lightweight *Publish/Subscribe* [49] messaging protocol , where multiple clients connect to a broker and subscribe or publish to topics they are interested in.

The **Publish / Subscribe (Pub/Sub)** paradigm allows many parties to communicate by means of asynchronous, many-to-many message exchanges, decoupling subscribers and publishers in space and time. Pub/Sub systems can be topic-based or content-based. In topic-based systems, each message is associated with a *Topic*, *Group*, or *Channel*. Publishers then publish their messages in a channel. Subscribers subscribe to the channel(s) they are interested in. If a system has the extra functionality that allows subscribers to provide a semantic filter on the content of messages it is designated as content-based.

The MQTT protocol, was introduced in 1999 by Dr Andy Stanford-Clark from IBM, and Arlen Nipper from Arcom. It was conceived for constrained devices and low-bandwidth with high-latency or unreliable networks. It is widely used in the emerging Machine-to-Machine (M2M) and IoT sectors, where an extremely simple messaging protocol and small client code are needed.

Many clients may subscribe and publish to the same topics and share messages, asynchronously. This characteristic makes it a good candidate even for a chat and message-based notification system as in the case of Facebook® Messenger application [88]. The MQTT solution is built on a *Client-Server* architecture, with a so light client that can stay even on microcontroller devices with small memory and processing capacity. A basic solution is made of a Broker (server side) and many clients. Clients should connect to a Broker and Subscribe to a Topic they are interested in. Any client connected to a Broker can Publish to a Topic and the Broker routes the message to all clients, which subscribed to the just published Topic. In such a way, clients can exchange messages in an asynchronous mode. Furthermore, a Broker can be bridged to another Broker, and the clients can still exchange messages, connecting to any of bridged Brokers.

In this project, the open source *Mosquitto* is used as MQTT broker [81] (server part) deployed on the Fog-Node. The client part is developed by using *Paho* libraries [59]. Both software are an Eclipse™ project. With a Broker deployed on the device in Layer 2, any clients deployed on each user's mobile device, can Pub/Sub to local Broker, by using the Wi-Fi access from any of the node in Layer 1 Figure 1.2. As the MQTT broker on Layer 2 and the end-user mobile devices are on the same Wireless Local Area Network (WLAN), the message latency is low, and messaging service is guaranteed even when the Internet connection is lost. At the same time, as the Broker on Layer 2 will be bridged with another Broker on the Cloud, Internet clients can join the notification-messaging system, directly connecting to the cloud Broker.

### *Localization*

The work in [117] presents four Generic Access Profile (GAP) roles for a BLE device. Only the *Broadcaster* and *Observer* GAP are used for the proximity location solution. A *Broadcaster* is a device that sends advertising packets, so it can be discovered by the *Observers*. This device can advertise but cannot be connected. An

*Observer* is a device that scans for *Broadcasters* and reports this information to an application. This device can only send scan requests but cannot be connected.

The Mist-Nodes of the proposed framework acts as *Broadcasters* and their BLE interfaces are sets as Scannable Undirect Mode, i.e. the node can be discovered by any other device but cannot get connected to it.

The end-user device acts as an *Observer* and forwards the *Broadcaster* information to the mobile application, particularly the Bluetooth RSSI, which will give the estimation of the distance between the mobile device and the Mist-Node. This last information will be used to localize the mobile device along the Mist-Nodes path.

### *Messaging*

The MQTT protocol is used for many tasks. It is used to pass parameters among the nodes, notification and short messages among the user-devices, and for centralized Orchestration of nodes. The Topic is crucial for message routing among MQTT clients. In the present use-case, the end-user mobile devices, Mist-Nodes, and eventually the Web applications that allow centralized management of the system, are MQTT clients, which publish and subscribe on Topics. It is not necessary any configuration about Topics to create it, it is enough to Publish or to Subscribe to it. Briefly, the Topics are a hierarchical structure, which uses a slash ('/') as a separator and the characters '+' and '#' as a wildcard.

When a client publishes a message "arrived" on a Topic "Node1/location/", and a message "call me" on a Topic '/Node1/msg/' the Topics are created. Any client that has subscribed to '/Node1/msg/' will receive the message "call me", while the clients which subscribed to '/Node1/#' will receive the messages "arrived" and "call me". The definitions of Topics are strictly application related. According to the example of Figure 1.3, the following scenario can be figured. A mobile application deployed on a user device, after joined the WLAN network through the Mist-Nodes, can publish the message "arrived" on Topic 'London-SSID/Client-ID/msg/'. The clients that subscribed to 'London-SSID/Client-ID/msg/'

will receive the message as notification of the presence of Client-ID at London-SSID, in real-time.

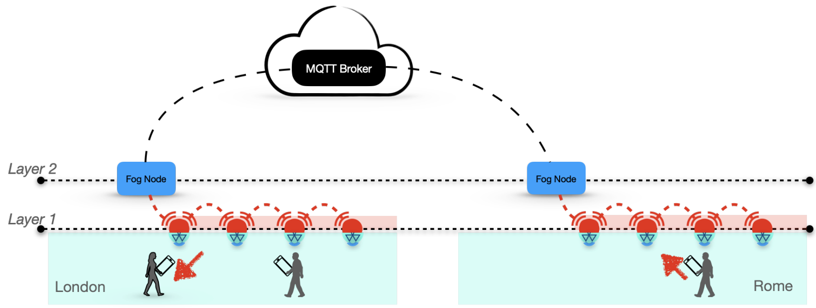



Figure 1.3: Multi-Branch company example

Another interesting mechanism is the bridging among the Brokers. When one Broker is bridged to another one, it is possible to set that all messages of a Topic, or a subset of Topics, published on any one of the Brokers, will be published on the others, too. Clients, which subscribe to a specific Topic, whenever they are connected to one Broker or to the other one, will receive the messages. As displayed in Figure 1.3, the notification of the presence of the Client-ID device, will be received by any Client connected to the Broker in London or to the Broker in Rome.

For example, if staff in the Rome office wanted to be notified when the company's accountant reached the London office, then the Rome staffing app should subscribe to a topic like "London/accountant."

Thus, when the accountant reaches the office in London, her/his application publishes any message to the topic "London/accountant." This action will allow the Fog-Nodes in Layers 1 (both in London and in Rome) to automatically route the new message to all subscribers of the topic, notifying them of her/his presence in the London office.

Indeed, the client applications need only to know the address of their local Broker. Actually, this is known because it is the same as the local Fog-Node at Layer 1, which must be known to the Root Fog-Node (since they are directly connected) and propagated to the mobile devices through the Mist-Nodes used for the Wi-Fi network access.

## 1.4 REMARKS

The solution supports rich services at the edge of the network with low latency requirements since *Fog-Nodes* are co-located with the user's smart devices, and data generated by these devices is processed much quicker than by a centralized cloud service.

The solution guarantee widely distributed and geographically-identifiable deployments, in sharp contrast to the more centralized cloud service, since the Fog-Nodes of the proposed framework acquire their geographical position and are discoverable by it.

The solution supports the collection and processing of data from different form factors acquired through multiple types of network communication capabilities. The nodes are heterogeneous in terms of processing power and storage capacity. Furthermore, each node has connection interfaces supporting multi-protocol communications and data collection. Indeed, both at Layer 1 and Layer 2, the nodes support Bluetooth, Wi-Fi, Inter-Integrated Circuit (I2C), and Serial Peripheral Interface (SPI) for serial communication with sensors and devices. Moreover, *Mist-Modes* and fog-nodes have different computational and storage capacities, empowering the Fog-Computing heterogeneity.

Using MQTT protocol, the communication among nodes is asynchronous but near real-time due to the message queued protocol, allowing real-time reaction.

The proposed model is easily vertically scalable by adding more Fog-node layers, when additional intermediate levels of processing are required, and horizontally by adding more inter-connected nodes on the same layer, both locally and remotely placed as in Figure 1.3.

From security point of view, *Secure Boot* and *Flash Encryption* are both present in the proposed nodes. *Secure boot* represents the technology addressed to assure that only the genuine code is running on the device chip, verifying the authenticity of code at each reset. *Flash Encryption* consists of encrypting the contents of flash memory so that, physical readout of it is not enough to recover most of the data.

# 2 HANDLING TRILLIONS OF EVENTS A DAY

## 2.1 SUMMARY

This chapter describes the work done in collaboration with French National Research Institute for Agriculture, Food and Environment (INRAE). The work has been defined in the context of the French National Research Agency Project ANR-19-LCV2-0011 Tiara, and French Government IDEX-ISITE Initiative 16-IDEX-0001 (CAP 20-25) and published in [3].

The use of a data streaming solution, capable of distributing millions of events per second to a huge number of connected objects, is described here. Following it is shown how we moved from the MQTT protocol to an event driven solution able to collect, in asynchronous way, millions of data per second. A use case is presented also. It consists of the scheduling and monitoring of autonomous agricultural robots.

## 2.2 INTRODUCTION

MQTT protocol is particularly suited for constrained clients due to the small footprint of MQTT client code. However, if the system requires to transmit millions of events per second and must be fault-tolerant, there are other technologies to consider.

In [5, 19] we experiment the Apache Kafka, an open-source distributed event streaming platform.

Apache Kafka initially conceived as a messaging queue based on the Pub/Sub paradigm is now an event streaming platform capable of handling trillions of events a day. The MQTT *Publisher* are called *Producers* in Kafka, the *Subscribers* are called *Consumers*, and the *Topics* are called *Channels*.

In collaboration with a team of INRAE, it was possible to experiment with a Fog-Computing architecture to collect and visualize spatiotemporal data produced by Autonomous Robots running at their experimental farm.

*Smart farming* is a move towards the adoption and the implementation of new advances in information technology to monitor and optimize agricultural business processes and overcome the current challenges of sustainable and healthy food production, defining new agroecology practices.

The deployment of affordable sensor technologies (e.g., odometry, Light Detection and Ranging (LiDAR), etc.) in farms, challenges traditional *Farm Management Information Systems (FMIS)*. Thus, FMIS must consider newly collected data features from data at rest and in motion, extracting insights. To improve field activities, such as precise Irrigation [34] and soil fertilization, as well as tasks scheduling and robots' tracking [91], must be analyzed these sensed data.

Robots have an essential role in supporting the agroecology transitions since they have a low impact on the environment (light, possibility to operate in the fleet) and can perform repetitive and accurate farming operations over a long period. With special equipment and combined with data acquisition and data processing technologies, robots can autonomously perform efficiently targeted tasks in the fields, e.g., within inter-cropping systems, while optimizing the use of resources and maintaining a high level of productivity.

Although many research works address smart farming, none has contributed with a fully-featured architecture design, which investigates the multiple layers and the different edges of the IoT ecosystem towards smart monitoring and scheduling of autonomous agricultural robots with scalable data processing. This lack has been put in evidence by some recent survey papers that focus on robots and IoT for Agriculture 4.0 ([1, 127, 138]). In particular, even though some client-server architectures have been proposed [45], they do not fully exploit all the insights extracted from the analysis of IoT data, the new communication protocols, and emerging computation architectures. Moreover, existing works do not address real-life scenarios but experimental farms configurations.

For this research work, were collected from farmers that work on the experimental farm in Montoldre (France) the users' requirements. This farm deployed more than 20 years ago, is an experimental platform used by INRAE to test agronomy and Information Technology research works.

A farming robot is an unmanned ground vehicle with sensors and actuators able to safely and autonomously perform one or several tasks on a farm field. Such a robot has a locomotion part connected to a navigation system and an agricultural part with mounted, semi-mounted, or towed implements. A farming robot can perform a specific task (e.g., weeding, harvesting, etc.) or be used as a tool carrier with various equipment combinations (e.g., tillage, seeding, fertilizing).

Autonomous agricultural robots scheduling is a logistical application to find the best allocation of robots for their agricultural tasks on the fields. This allocation has to be done based on:

1. compatibility of a robot and its equipment with an agricultural task;

2. the temporal availability of a robot and its associated equipment;

3. the meteorological conditions compatibility with an agricultural task.

When these conditions are satisfied, the output of the scheduling algorithm is a set of predefined trajectories that the robot must follow at the scheduled time.

Typically, the design of a reference trajectory is based on simulation models [29]. Other empirical approaches may be based on analysis of past robot trajectories. However, some problems may arise during the execution of an agricultural task (e.g., a physical object blocking the robot's movement, heavy rain, or snowfall). As a result, the robot's planned trajectory needs to be updated. Thus, an online reprogramming computation service must be run to find (when possible) the best alternative trajectory.

The following events may trigger a re-scheduling of planned tasks:

- *Robot fault*: since robots operate in rural environments frequently under challenging weather conditions and on topologically harsh grounds (e.g., ground elevation, rocky floors), they may experience technical problems.

- *Delay alert*: robots are programmed to follow a scheduled trajectory, but they are also allowed to automatically re-adjust their trajectories to avoid obstacles (e.g., animals, tree branches, rocks) [130]; any trajectory re-adjustment results in delays.

- *Meteorological alert*: weather conditions (like heavy rains, thunderstorms, high winds) can badly impact a scheduled trajectory. For instance, a spraying task cannot be performed during strong wind, and in such a condition, a robot must stop its work.

The previous events may require the rescheduling of tasks due to spatiotemporal and contextual issues.

Spatiotemporal issues refer to the lack of respect for the spatiotemporal constraints associated with predefined trajectories for one or several robots. These issues occur due to robot(s)' deficiency (i.e., Robot fault event) or an event external to a robot but associated with a particular spatial or temporal point (i.e., Delay alert event).

Contextual issues refer to problems caused by a phenomenon that cannot be associated with a particular spatiotemporal point but is valid for a set of plots or technical operations (i.e., Meteorological alert).

A *smart farm information system* involves the following stakeholders.

- *Farmers*: they have to monitor in real-time the trajectories of robots on their plots. They must also be alerted in real-time when problems occur.

- *Scheduling experts*: they are users with logistic skills who set up scheduling algorithms to provide the most efficient reference trajectories for all plots, robots, and activities. The system must provide the details on all executed scheduling tasks in the past and allow them to explore and analyze this data. They typically react in time windows of different lengths (e.g., 1 hour for re-scheduling operations, a few days for the reconfiguration of the scheduling algorithms).

- *Mechatronics engineers*: they off-line analyze the behavior of robots to update and adjust robots settings.

The deployment of IoT technologies in farms generates Big Data [44, 103]. For this reason, agricultural autonomous robots scheduling systems must be able to handle data characterized by the well-known three V's, namely,

- *Velocity*, since sensors and network devices produce continuous data during the whole day of the work of the robot. For instance, each robot emits 200 odometry records per second that are used to analyze its mechanical behavior experts. Also, the monitoring of robot faults is performed in real-time.

- *Volume* -since a lot of data is generated with high frequency, massive volumes of data are produced, which have to be stored and analyzed.

- *Variety*, since the data is in different formats, structured (e.g., robot trajectories), semi-structured (e.g., GeoJSON produced by meteorological sensors), and unstructured (e.g., photos and videos taken by robots, upon a request from a farmer, to take appropriate corrective trajectory actions).

Consequently, suitable software applications for analyzing historical data, processing data in real-time, and handling different data formats need.

The scheduling of the fleet of robots must support the following types of queries:

- *Analytical queries* explore historical data. Typically include Online Analytical Processing (OLAP) queries, and Machine Learning algorithms, to improve the scheduling algorithm. Logistic experts could be interested in the duration (e.g., min, max, average) of trajectories across task types, robots, weather conditions, and farms. These queries involve data related to the different farms, and usually, they are computed offline.

- *Continuous queries* run over streams of data. Continuous queries are triggered automatically with a fixed frequency. These queries involve data of a single farm. An example of a continuous query is: "Calculate every minute, the accumulated delay for each robot, over the last 10 minutes".

- *Create, Read, Update, Delete (CRUD) queries* run over dictionary data (e.g., reference trajectories). These queries may use data from different farms to check the availability of robots. An example of a CRUD query is: "Update the reference trajectory due to a re-scheduling operation."

Finally, these queries feature different time computation constraints ranging from hard real-time ( e.g., hard real-time time computation for robot fault queries) to offline (e.g., OLAP queries).

The experimented system took advantage of all the economic benefits and flexibility of the cloud technologies, combining them with the advantages of a Fog-Computing architecture.

On *Fog-Nodes*, an intermediary effective communication network layer *Access network* between field robots and the cloud was deployed. Commonly, robots and wireless sensors cannot be connected to the Internet. Indeed, depending on the location of the farm, cellular networks might not have sufficient coverage to support data transmission. Therefore, they cannot be connected directly to the cloud. Moreover, the *Access network* must support dynamic configurations, since robots are mobile, and by consequence their dynamic location makes it difficult to communicate to the network. One of the candidate technologies to be used for the *Access network* is *Wi-Fi*.

The *Access network* must also offer enough bandwidth and high reliability to be able to transmit critical real-time data from the robots to the farmer (such as video and images necessary for the remote control guidance of robots in case of unexpected or faulty behavior of robots). Communications should also offer low latency links especially for the real-time commands sent by the farmer to the robots. Therefore, the *Wi-Fi* network should be well designed, i.e. with a good coverage and sufficient number of access points, in a way to guarantee these requirements in terms of network performance.

A deployed *control station*, managed by the farmer, plays the role of a gateway for the *Access network* to offer Internet connection to the robots. This control station is reachable through the *Wi-Fi* network, and is connected to the cloud utilizing classical protocols such as *ADSL*, fiber optics, or any other type of high speed connection. Therefore, the system must be deployed as a *Fog-Computing* one, where robots represent the *End-Devices*, and the control station of the farmer represents the *Fog-Node*.

It has been proved that the Lambda architecture [85] allows to efficiently process streaming data and static (warehouse-like) data, e.g., [66, 96]. For this reason, to meet the requirements described above, a skeleton of a new architecture called *LambdAgrIoT* was proposed, as shown in Figure 2.1.

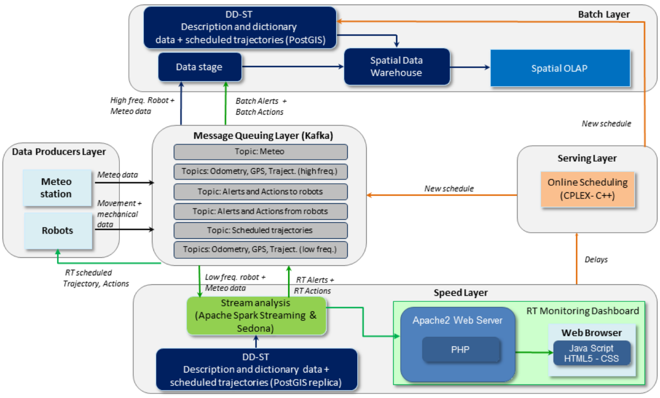



Figure 2.1: Components of the *LambdAgrIoT* architecture from [19]

*LambdAgrIoT* is composed of the following five layers.

- *Data producers layer* represents data sources deployed in the field, i.e., robots and meteorological stations. They produce data sent to the *Events Streaming* layer.

- *Events Streaming layer* is in charge of managing data (messages) that are exchanged by different layers. Robots also receive data from this layer.

- *Speed analysis layer* provides tools for analyses of data in real-time. The layer is deployed in each farm.

- *Serving layer* supports scheduling robots and their tasks.

- *Batch analysis layer* centralizes all data from all farms employing the management and the analysis of historical data. Notice that this layer is in a very early stage of development and is part of the closest future work.

These layers are deployed according to a *Fog-Computing* Architecture.

These layers have been deployed in our *Montoldre* experimental farm (as a part of the Batch layer that we are finalizing). This actual deployment confirms the feasibility of our approach, and at the same time, we are conscious that more advanced experiments should be conducted at a large scale.

## 2.3 LAMBDAGRIOT LAYERS

Below are detailed the main features of each layer with the configuration of the machines used for the implementations. Moreover, are reported the results of tests performed.

### *Data producers layer*

As mentioned earlier, the two data producers are weather stations and robots. The station delivers multiple weather data measurements, such as temperature, humidity, dew point, wind speed and direction, wind chill, barometric pressure, rainfall, rain rate, UV and solar radiation data, using radiofrequency and Local Area Network (LAN) converter. To a specific topic at the *Events streaming layer* are published these data. Both the *Speed analysis layer* and the *Batch layer*, using a stream pattern, consume weather data.

Robots run on *Robot Operating System (ROS)*, which is an open-source platform for robot software development. It provides a set of functionalities for hardware abstraction, low-level device control, and messaging. It comes with external libraries, such as *rospy* and *roscpp*. *Python* scripts implement the odometry and Global Positioning System (GPS) data acquisition module. These data are acquired 200 times per second and then sent via a socket to the *Events streaming layer*. Therefore, the *Python* code represents a *data producer* for the distributed streaming platform deployed at the *Events streaming layer*. The socket is also always listening for new reference trajectories, turning into a *data consumer* whenever a rescheduling operation happens.

**Deployment:** Actually, one *Davis Vantage Pro 2* wireless weather station have been installed in our farm and two *Adap2e* autonomous robots developed by INRAE [37] were used. The usage of two robots corresponds to a realistic scenario since fleets of a bigger number of robots are not yet a matter of fact in the French agriculture context.

*Events streaming layer*

The proposed system requires (1) a scalable and fault-tolerant message delivering in real-time, as well as (2) the Pub/Sub and the stream interaction patterns. Therefore, we use *Apache Kafka*. The choice is also motivated by the recent performance analysis of *Kafka* [41, 63], demonstrating high data ingestion rates.

In this architecture, data producers (i.e., robots and the weather station) publish messages to specific *kafka channel*. A *kafka channel* is a logical grouping for messages for all the consumers subscribing to that topic. Then were created the following *kafka channel*:

- *Weather topic*: it stores data from the weather station. This topic is consumed by the *Speed analysis layer* for real-time analysis and by the *Batch layer* for storing historical data.

- *Odometry low-frequency topic* and *GPS low-frequency topic*: they include data sent by a robot every second. These data are used by the *Speed layer* for real-time analysis. *Odometry low-frequency topic* represents mechatronic data of the robot. GPS low-frequency topic represents the real-time trajectory of the robot by the triplet <*GPS-coordinates, timestamp, idrobot, speed*>.

- *Odometry high-frequency topic* and *GPS high-frequency topic*: they include data sent by robots every millisecond and are used exclusively by the *Batch layer*.

- *Scheduled trajectory topic*: it includes reference trajectories re-scheduled by the algorithm at the *Serving layer*.

- *Alerts and actions topic*: it includes alerts computed by the *Speed layer* and the corrective actions (automatically determined by the system or by the farmer). Therefore, data producers for *Alerts* and *Actions* are deployed in the *Speed* *layer*. Consumers are *Batch layer* and the robots, which retrieve from corresponding sub-topics actions they must perform.

### *Speed layer*

Three modules compose the *Speed layer*, namely: the *Stream Analysis Module*, *DD-ST Module*, and *RT Monitoring Dashboard Module*.

THE STREAM ANALYSIS MODULE consumes data streams produced by *data producers* (robots, weather station), and published in specific *kafka channel*. The analysis of data streams allows: (1) tracking work progress in real-time (calculate delays if any) and (2) signaling alerts and recommending corrective actions. This module requires scalable in-memory computations. For that purpose, we use *Apache Spark* and *Apache Spark Streaming* [136, 137]. In order to handle spatial data and spatial operators we use *Apache Sedona*. *Sedona*, *Spark*, and *Spark Streaming* process data coming from *Kafka*.

The implementation of *warning generation* (i.e., robot failure alerts and delay alerts) and corrective action management (i.e., schedule update and trajectory update) were managed at the *Speed* layer for multiple reasons such as minimal maintenance of the robots' software and to allow correlation analysis with other events.

***Description and Dictionary data, and Scheduled Trajectories (DD-ST)*** is a transactional spatial database that aims to store data used by the system. It is deployed on top of *PostgreSQL* relational database, implementing the *PostGIS* spatial extension. The database stores, among others, reference trajectories, geometries of farms' plots, and multiple robot data. The *Speed analysis layer* also used these data to resolve continuous queries and compute alerts. Since the communication network between the farm and the cloud is not reliable (typically an *Asymmetric Digital Subscriber Line (ADSL)* connection), it is not appropriate to query the cloud database to meet low-latency requirements. Therefore, we install a local replica of the cloud database on the farm. The database is made available as a database as a service.

***RT Monitoring Dashboard Module*** is used to analyze and visualize data computed in the *Stream analysis module*. The real-time data visualization application was designed to provide a quick and simple way to notify the farmers about alerts and provide a visual analysis tool for the actual trajectories of robots. The software was developed as a web application to allow multi-user interaction and be multi-platform. It is executed partially on the client-side (by any web browser) and partially on the server-side by the *Apache web server*. The user does not need to install any native application on his device but only access the web page served by the local web server installed on the edge device in the farm. Through *PHP* scripts, the web server can directly consume *Kafka* data streams produced by *Spark* analysis. Furthermore, the same *PHP* script, employing Server-Sent Events (SSE) technologies, forwards *Kafka* streaming data to any connected browser in real-time. The SSE allows the browser to get a notification for each new data available without multiple Hypertext Transfer Protocol (HTTP) requests. In this way, the web page automatically updates whenever the data change.

The web interface allows farmers to visualize all information about the execution of the scheduled tasks. It is composed on three main panels (cf., Figure 2.2): (1) *Interactive and dynamic Gantt*, which is responsible for the visualization of the advancement of each robot in its task; (2) *Dynamic graphical charts*, which visualize the numerical data (weather, odometry, and speed); (3) *Interactive and dynamic map*, which visualizes the real-time trajectories of robots and their reference trajectories. The reference one is entirely drawn on the map from the beginning using the yellow color, while the real trajectory is drawn in real-time using the red color. Due to the color blending on the map, the trajectory will look orange when both the trajectories sign match, but we will have a Red trajectory sign when the real-time one does not match the reference one.

As suggested in [84], this geovisualization system uses multiple coordinate views, which consists of providing several synchronized visual representations of the same data. When the farmer clicks on a particular line in the Gantt chart, the map will be centered and zoomed into the corresponding plot, where the robot is working. Although the literature reports various studies about the Gantt chart linked to graphical visualizations, no work proposes the synchronization with maps. The visual analytics mantra [28], "Pan, zoom, and filter first then detail on-demand" inspired this coordination. It allows the farmer to get an overview of the planned activities using the Gantt chart (*First Overview*,) where is shown the time dimension for the different robots, but the spatial and numerical details of the real-time trajectories are not. Then, using the Gantt, the farmer can zoom in on the trajectory of a particular robot (filter) and see information about its work (details on-demand) using the map.


Figure 2.2: RT Monitoring Dashboard Module example in the current deployment from [19]

**Deployment:** Spark Streaming (version 2.3.3) running on java 1.8 installed on a machine with Windows (version 10), as the operating system, was used for this layer. The machine had 32 GB of RAM and a 64-bit quad-core i5 processor. For the system testing, continuous query produced alerts on the web interface The query estimated the average speed of robots based on its odometry data provided in a window of the last 30 seconds. The aggregation is calculated incrementally by Spark (using micro-batches). Each time new data arrives, a micro-batch updates the aggregation calculation. Each micro-batch takes 0.3 seconds revealing the real-time feasibility of our approach. A video showing the functioning of the web interface is shown at this URL https://youtu.be/e52gNgseQFg.

*Serving layer*

The scheduling service has to satisfy the farmers' requests respecting the constraints of robots configuration. It is modeled as a mixed integer linear program (MILP) that can be solved by linear program solvers like *CPLEX* with *C++* [5, 19]. As input, this program needs, for each robot and equipment, the date it will be available, and as output, it provides the subsequent reference trajectory of each robot (cf. Figure 2.3).

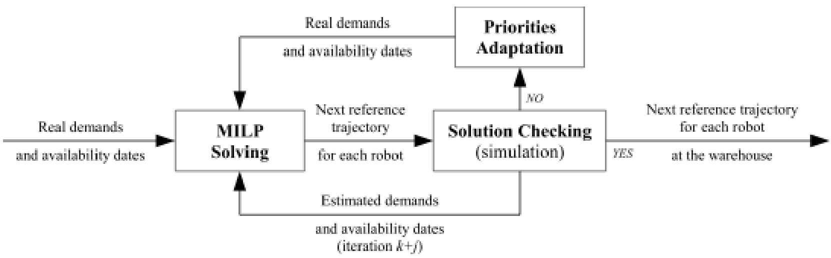



Figure 2.3: Finding next reference trajectories with online scheduling from [19]

Using linear programming or Petri net, solving the entire scheduling problem at once is difficult as it requires a long computational time. Therefore, an iterative approach was chosen by adding some constraints to the model to prevent most of the blocking situations.

**Deployment:** Our online scheduling algorithm was implemented on a computer with an Intel Core i5-8400 @ 2.80GHz × 6 CPU, 15 GB RAM, Ubuntu 20.04.2 LTS 64-bit operating system, and using CPLEX version 12.10.0.0 for linear programming resolution. We tested realistic farm scenarios, with six plots, three to six pieces of equipment, three to six robots, and three to eight configurations. The algorithm execution times for these scenarios are less than one second giving us a good initial point on the solution for online scheduling.

*Network Architecture*

A Wi-Fi network allows the farmer to have a permanent, low-budget, and operator-independent communication network to control and supervise the robots. Figure 2.4 and Figure 2.5 show the *Fog-Computing* architecture. Robots represent the *End-Devices* and the farmer control station represents the *Fog-Node* connected to the *Cloud*. In our case, the *Fog-Node* plays a crucial role. It offers an Internet connection to the robots, offloads processing from the robots, and most importantly, allows real-time interaction between the farmer and the robots. The proximity and the quality of the Wi-Fi network must allow real-time interaction that guarantees the application requirements (i.e., low delay, high data rate, low data loss).

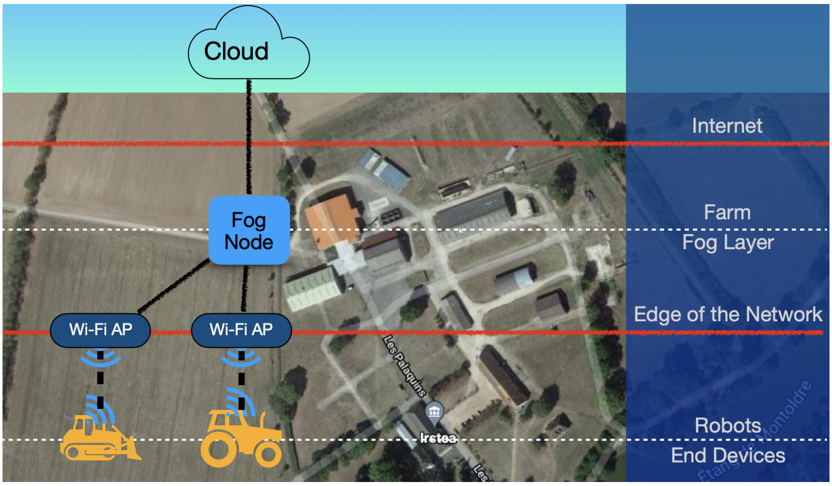



Figure 2.4: The *LambdAgrIoT* network architecture

In this architecture, we can distinguish two types of networks: *Access network* and *Collect network* (Figure 2.5). The *Access network* ensures quality communication links between the robots and the farmer (Fog node) through the Wi-Fi network. The *Collect network* offers a communication link between the farmer and the cloud servers for further analysis and storage through a broadband connection (e.g., ADSL, Fiber, Cable).

## 2.4 REMARKS

This work allowed the experimenting with the gathering of heterogeneous data from multiple sources using open source tools. One significant accomplishment was leveraging the Pub/Sub

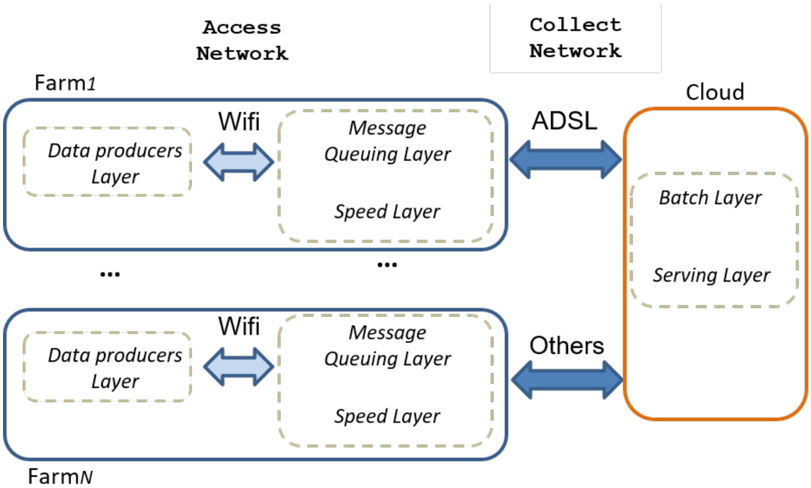



Figure 2.5: *LambAgrIoT* network and modules mapping from [19]

paradigm to decouple data producers and consumers, in space and time, through the use of the Kafka event streaming platform. The proposed solution analyzed, visually and consistently, data from sources producing data at different rates, along with data from database queries. The architecture had success even for potentially large and fast streams of events. Another remarkable accomplishment has been the experience with multi coordinate views technics and the visual analysis employing color blending strategies.

# 3 COMPUTATION OFFLOADING

## 3.1 SUMMARY

This chapter discusses the use of the Pub/Sub messaging solution to distribute computational functions among compute nodes in a wireless network. The paper emphasizes the concept of asynchronous communication, proposing a framework that, without orchestration, is capable of autonomously distributing computation over a network of nodes by combining the concepts of *Collaboration* and *Competition*.

## 3.2 INTRODUCTION

Presented at the "Internet and Distributed Computing Systems conference" in 2019 (IDCS 2019) [16], this work employs the MQTT protocol, previously described in Chapter 1, to accomplish a Remote Function Invocation among the nodes of a Fog-Mist Computing architecture.

MQTT client libraries exist for multiple programming languages and various constrained devices. The MQTT *Pub/Sub* message exchange protocol provides many-to-many message distribution and decoupling of applications.

Furthermore, the MQTT protocol has Quality of Service (QoS) management. With its three levels, it is possible to choose "At most once," "At least once," and "Exactly once" for the message delivery. An additional mechanism to notify interested parties when an abnormal disconnection occurs makes it resilient.

## 3.3 THE COMPUTATION OFFLOADING ARCHITECTURE

**Computation offloading** refers to the transfer of computational tasks to a remote device or platform, and the following Figure 3.1 depicts the architecture designed to experiment with the computation offloading solution utilizing the MQTT protocol.

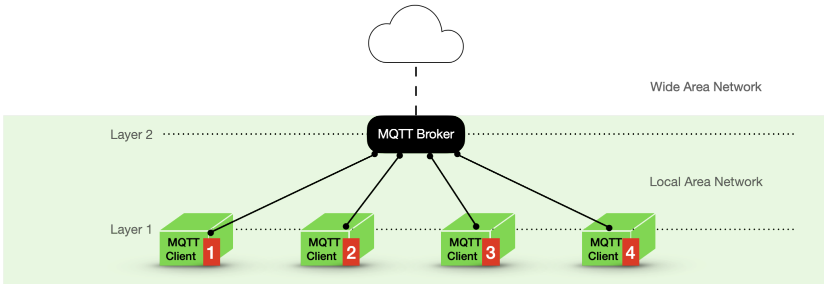



Figure 3.1: A Fog-Mist computing architecture to experiment computation offloading solution by means of the MQTT

In this case, the *MQTT Broker* resides on a local connected *Fog-Node* at Layer 2, which by definition has more computational and storage resources than the *Mist-Nodes* placed on Layer 1.

**The *Mist-Nodes*** are all on the same Local Network, co-located with the end devices. Data generated from end-devices are processed by the *Mist-Nodes* faster than by a centralized cloud device, satisfying the requirement of low latency necessary for devices that must react to stimuli quickly.

**The *MQTT Clients***, residing on each *Mist-Node*, can *Subscribe* and *Publish* to any *Topics* on the local broker.

An essential part of the strategy relies on the ***Topic*** structure of the MQTT protocol. As previously seen, it is similar to a file path. Since the *Broker* has a root '/' *Topic*, any *Publisher* or *Subscriber* can build a hierarchy on it. For example: "/Request/*functionA*" and "/Request/*functionB*" are two *Topics*. It is not necessary to create a *Topic* before. A *Topic* is created whenever a client *Subscribes* or *Publishes* it the first time. By use of the wildcard character '#', nodes which *Subscribes* to the *Topic* "/Request/#" will receive the messages *Published* on both the sub-*Topics* "*functionA*" and "*functionB*." Any Client which *Subscribes* to *Topic* "/Request/*functionA*", instead, will only receive the messages *Published* on this *Topic*.

## 3.4 RELATED WORK

Calice et al. in[24] investigate multiple offloading strategies concerning both computation time and energy consumption, related to heterogeneous devices forming mobile-to-mobile opportunistic computing. Le Minh et al. in[78] present a novel distributed execution model that both optimizes mobile application's executions in terms of performance and energy efficiency, and expands mobile device's hardware capability. In[77], Le Minh et al. introduce a new software framework enabling routing Remote Procedure Call (RPC) architecture on multiple groups device-to-device networks, arguing that "the RPC-based or message queue-based techniques are obsolete or unwieldy for mobile platforms".

The cited works, mainly face the challenge of heterogeneous hardware and try to find solution for two of the main issues, inadequate Internet connection and services discovery. By bringing computational power at the edge of the network, the Fog-Mist Architecture does not require a reliable Internet connection. Furthermore, the nature of proposed Fog-Mist Computing Framework, made up of homogeneous firmware *Mist-Nodes*, will avoid the need to know which function is available and by which resources. Then, solutions based on this architecture can benefit from a more straightforward method to remotely invoke a function from another local node, which not necessarily has the same hardware, but that has mainly the same implemented functions.

Thus, this work proposes a computation offloading solution, to achieve a Remote Function Invocation (RFI) rather than a RPC or a Remote Method Invocation (RMI), to lower the latency time for the mobile devices, which are connected through Fog-Mist Computing Architecture. Other than from previously cited works, which deal with a higher degree of difficulty, this proposal takes all the advantage of previously mentioned Fog-Mist architecture, mastering many of the issues faced in the related works.

It is assumed a two-step paradigm based on *Collaboration* and *Competition*, where all node are same.

Finally, although the proposed solution has at least one MQTT Broker caring about the routing of messages, it does not have any *Orchestration* or client-server model for the computation offloading.

## 3.5 COMPUTATION OFFLOADING WITH MQTT PROTOCOL

In most cases, the *Mist-Nodes* are made up of the same hardware having the same firmware supplying the same functionalities.

In this solution, whenever a node needs more computational resources, it enquires other sibling nodes for collaboration, starting a competition.

Here are named as *Demander* the nodes that require the collaboration, and *Offerer* the nodes that are available to offer their cooperation. Any node can be a *Demander* or an *Offerer* at any time, without any *Orchestration*.

### *The Demander*

A *Mist-Node* requiring the execution of a remote task, or needing some information from other local devices, can *Publish* a message on the local MQTT Broker to demand a *Collaboration* to other siblings, becoming a *Demander*. It can wait for an answer while it is too busy to elaborate the task by itself. Once the reply is available to the *Broker*, the *Demander* is notified and can use the results. Instead, if the *Demander* acquires resources to elaborate the answer by itself before it is available from other nodes, then it can proceed with elaboration, ignoring the possible response.

### *The Offerer*

In this example, every local *Mist-Node* subscribes to the *Topic* "/Request/#." Thus, whenever a *Demander* publishes a message to any *Topic* beneath this one, for example "/Request/*functionA*," the other nodes will receive the message. If the node has available resources and is in an idle state, it can decide to execute the required elaboration, becoming so an *Offerer*.

An *Offerer* initially unsubscribes to "/Request/#" and subscribe to the specific topic, in this example it is "/Request/*functionA*". The first *Offerer* that finishes its elaboration, will publish the computed answer on the "/Request/*functionA*" topic and unsubscribe from it. This operation will communicate to all involved nodes the fulfillment of the task. The other *Offerers* can then abort their execution, and the *Demander* can take its result. When ready to accept other requests, the nodes will subscribe to "/Request/#" again.

Following the previous examples, the *Message*, when published by a *Demander*, has to be published on a sub-topic of root topic: "/Request/," named as the *Function* required, and the message content must contain data or parameters that have to be computed by the *Offerer*.

For example, a *Demander* which asking for elaboration of *functionA (a,b)*, will publish to the topic: "/Request/*functionA*" with the message content: "a,b".

To summarize, when a new request rises from a *Demander*, all subscribers to the topic "/Request/#" will be notified. The *Offerer* that faster completes the elaboration, will publish the message containing the answer to the specific *Topic* (e.g, "/Request/*functionA*") and the *Demander* will be notified together with all the other competing *Offerers*. While the *Demander* can take and use the answer, the other *Offerers* can abort their elaboration, subscribing again to "/Request/#", ready to receive notification of new requests.

Figure 3.2 schematize the example above.

## 3.6 THE UNDERLYING PARADIGM OF COLLABORATION AND COMPETITION

The *Computation Offloading* starts when a *Demander Publishes* a request. A phase of *Collaboration* follows this event, where all the other nodes can spontaneously decide to offer their collaboration. The decision will be based on their busy state and their capacity at the moment of request. No *Orchestration* or *Arbitration* is required.

To the Collaboration phase, follows a *Competition* phase, where all the *Offerers* compete. The winner will be the fastest *Node* to *Publish* an answer.

## 3.7 REMARKS

The proposed solution is easy to deploy, just leveraging the MQTT protocol. Being based on a *Collaborative* and *Competitive* paradigm, it implies that an *Orchestration* or a service discovery are not required. The low latency requirement and the unreliability of

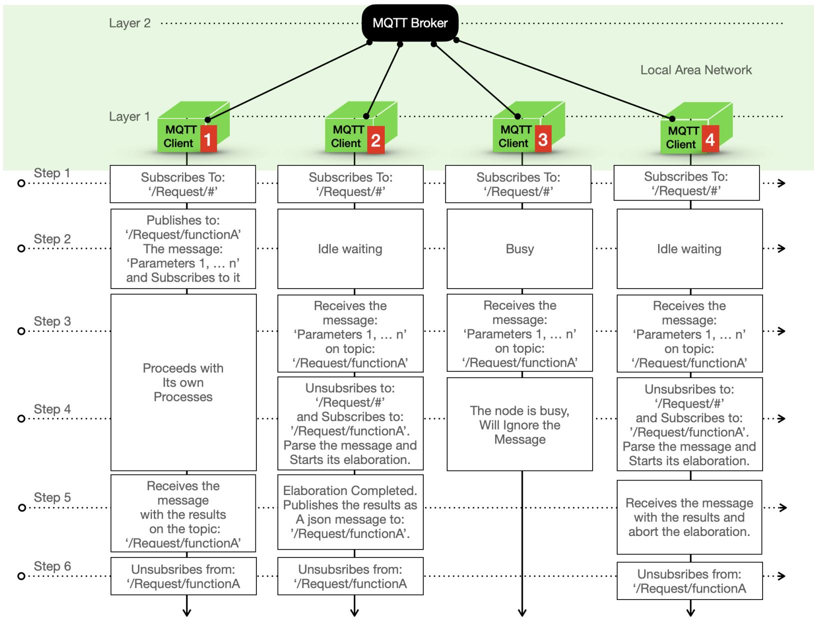



Figure 3.2: MQTT Messages Sequences for the Remote Function Invocation

the Internet connection are featured by Fog-Mist Architecture, while the messaging protocol itself solves QoS issues. This work was conceived to exploit the increasing adoption of the Fog-Computing paradigm and thePub/Sub protocol, even in different domains. Moreover, a *Fog Node*, with more computational resources, could transparently become an *Offerer*. It could offer, for example, inference for a Deep Learning Model, or in general, a heavy maths computation, to the simpler *Mist Nodes*, moving the Artificial Intelligence (AI) at the Edge.

# 4 SMART OBJECTS AS SENSORS NETWORK BASED ON FOG-COMPUTING

## 4.1 SUMMARY

In this chapter, a framework for collecting data from wearable devices is proposed. The data sources are mobile, so location plays a key role in the proposed use case for testing the framework. Through a Mist-Fog Computing architecture, spatio-temporal data is collected from specially designed wearable devices and displayed on mobile devices to preserve worker safety.

## 4.2 INTRODUCTION

This chapter presents the work described in [18] that demonstrates the Fog-Computing paradigm in collecting sensor data and locations. Furthermore, the proposed solution uses the MQTT protocol to visualize sensor data in real-time by a web application.

According to the European Agency for Safety and Health at Work [55], occupational accidents currently still represent a high cost for companies and, more importantly, a much higher cost for victims and their families. Therefore, the research investigated a use case where innovative smart Personal Protective Equipment (PPE) represents mobile end-devices of a Fog-Computing architecture.

The European Occupational Health and Safety (OSH) recommendations give to PPE the highest preventive effectiveness. The European Economic Community (EEC) directive on its use affirms that PPE must be used whenever risks cannot be avoided or sufficiently limited by collective protection procedures and technical means [51].

Despite this awareness, the number of workplace accidents that seriously harm people remains high. Although the success of some preventive measures certainly depends on the right choice of PPE and its proper fit, it is also important that it be worn at all times, and this requirement is often disregarded, as a survey by Kimberly-Clark [74] shows. The survey states that 82% of safety professionals say that workers in their organizations fail to wear the required PPE.

The Kimberly-Clark study points out that it is critical to make sure that PPE is worn all the time. In addition, safety precautions are even more important when dealing with individuals who work alone, without supervision, and isolated from other workers.

Apart from the same dangers as everyone else, *Lone Workers* are at greater risk, since they have no one to ask for help in the event of an accident. Therefore, additional systems are needed to keep in touch with workers and monitor them remotely. In particular, innovative solutions are needed to trigger alarms automatically or manually, know workers' location and detect that they have returned to their base once they have completed their task.

A recent discussion paper [52] by the European Agency for Safety and Health at Work describes the smart PPE as smart protection for the future. Although a European standardization definition is still only a proposal, it is common to define *smart PPE* as 'the combination of traditional PPE with intelligent elements'. As shown in Figure 4.1, most smart elements are electronic.

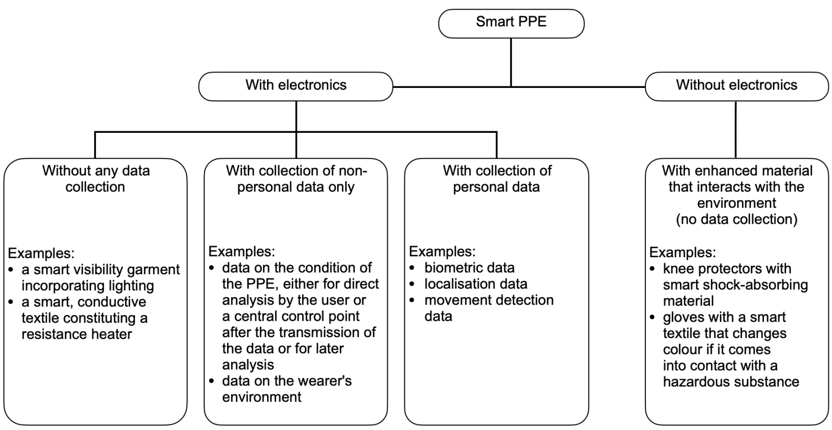



Figure 4.1: A proposal to classify a smart PPE, according to composition and data collection capabilities, as described in [52].

Most of them collect data and transmit it in near real-time when continuous remote monitoring of workers' status is required. These solutions become less attractive to employers and are even unsuitable for some outdoor workplaces if they are expensive or complicated to install.

Smart PPEs should promote their acceptance, and [52] emphasises that clear safety benefits and low cost are essential requirements. Since good PPE is PPE that is used all the time, smart PPE should be used seamlessly in workplaces. Based on the above considerations, the research direction was to identify solutions that let the safety professionals verify the continuous use of PPE and offer the possibility of a prompt rescue in the case of an accident.

To achieve these goals, several requirements have been identified that affect different design phases. In particular, solutions are needed that are based on a simple infrastructure and are cost-effective at the time of acquisition to make the solution attractive and feasible from the employer's perspective. This means that the electronics that transform PPE into smart PPE must have minimal impact on traditional PPE. In addition, the way the data is collected should not worry employees.

Considering the effectiveness of a Fog-Computing infrastructure, this chapter proposes a practical and affordable solution based on dedicated hardware to transform traditional PPE into smart PPE and implement the required wearable sensors. Moreover, it can also be deployed on an open and large worksite, limiting the use of cloud resources to secondary tasks, making Internet connectivity optional, and the whole system tolerant to unreliable Internet connectivity.

## 4.3 RELATED WORK

This section lists works that address solutions for worker safety through smart PPE and wireless sensors. They were studied as a starting point for the proposed improved solution.

In [112], Shabina proposes a *smart helmet* equipped with sensors to monitor the working environment of underground miners.

In [72], the authors propose a dedicated wireless sensor to prevent fatal accidents with vehicles on construction sites. With

[80], Haibo Li suggests using Radio-Frequency Identification (RFID) tags and readers to alert dockworkers when they are in a dangerous zone, such as the work area of a crane.

In [30], the authors propose a prototype for a wearable tactile sensing system that alerts construction workers to potential hazards. Notifications are sent via a tactile language system.

In [70], the authors acknowledge the increasing use of IoT sensors for worker safety and propose a framework for collecting and managing these devices. They also describe a sensor-equipped smart helmet developed to validate their idea.

A system detecting anomalous situations is proposed in [56]. It uses waist-mounted and environmental sensors alerting workers in case of hazard possibilities. Moreover communicates their position only in case of an accident, preserving their privacy during ordinary working situations.

In [134], the authors propose multiple wearable sensors to monitor environmental and workers' physiological parameters. The wearable sensors can communicate among them and transmit data to a gateway. Once dangerous situations are detected, the sensor node provides users and the remote site with a notification and a warning mechanism.

The use of a neural network solution to detect if workers are wearing helmets during their activities is proposed in [83, 85].

In [87], the authors describe a smart IoT device developed to detect human falls and monitor workers' vital parameters, by sending an Short Message Service (SMS) notification for immediate aid when such signals are indicating an accident.

In [97], the authors propose wearable devices and *Edge-Devices* to train a neural network model to detect worker's falls by using Deep Learning video cameras. They claim an achieved falls detection performance above 94%.

In [69], Jayasree and Kumar propose a helmet embedded with an accelerometer and a gyroscope sensor to monitor construction workers' physical conditions and send notifications to the contractor via mobile.

## 4.4 PRELIMINARIES

The research proposes a comprehensive solution providing continuous monitoring of PPEs usage, and workers' localization when an accident occurs, for a quicker rescue. Made up of low-cost devices embedded in a *Fog-Computing* architecture, it leverages the most popular IoT technologies.

The proposed solution, having low recurring maintenance and connectivity costs, encourages its deployment, even in large and temporary workplaces. The worker's privacy is guaranteed, and it has little impact on their usual activities.

Figure 4.2 depicts a typical Edge-Computing architecture, having a Wi-Fi area as the Edge of the Network for all the wireless connected devices, and some *Edge-Devices* between the end-user devices and the Wide Area Network (WAN).

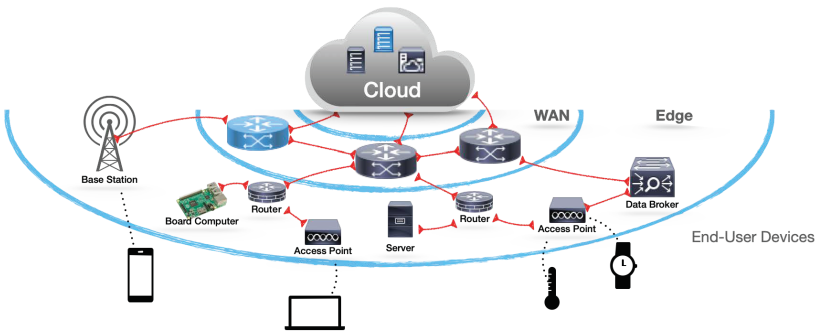



Figure 4.2: An Edge-Computing architecture scheme from [18].

Using the MQTT protocol [88] the solution assumes that a *Topic* identifies the worksite and subtopics identify its workers. Then "$Site/User_i$" addresses the $i$th worker of the site. Furthermore, each worker has a subtopic "status", then "$Site/User_i/status$" is the *Topic* for the situation of $i$th worker on the site.

Since the MQTT protocol allows the use of wildcards as a filtering possibility, clients interested in all users' status, working on *Site*, can subscribe to the *Topic* "*Site/#*" being notified about all the messages published on every *Topic* below "*Site*", i.e., all messages about the status of all users at that site (Figure 4.3).

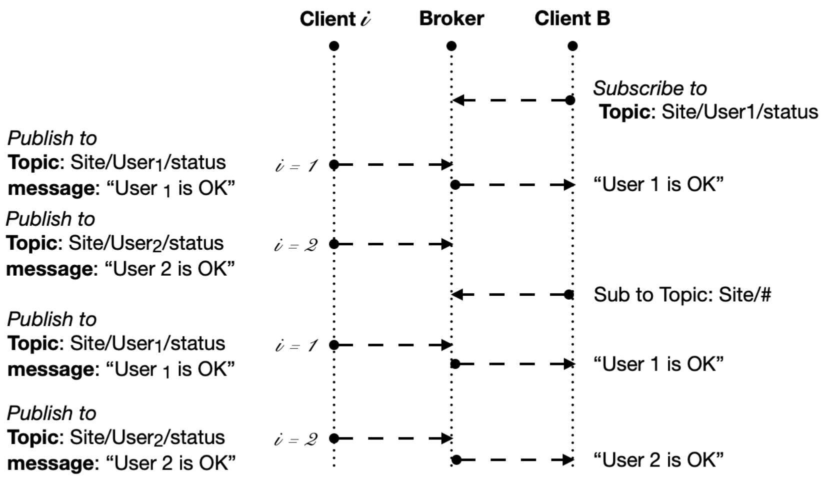



Figure 4.3: Publishers and subscribers, examples without and with wildcard from [18].

Clients interested in the status of a given $User_i$ can subscribe to the specific *Topic* "$Site/User_i/status$" and receive the messages strictly related to that worker.

A *Broker* can work stand alone or *Bridged* with another *Broker*. In the latter is possible have a load-balanced, a mirrored server or the access to the same *Topics* by two different addresses Figure 4.4).

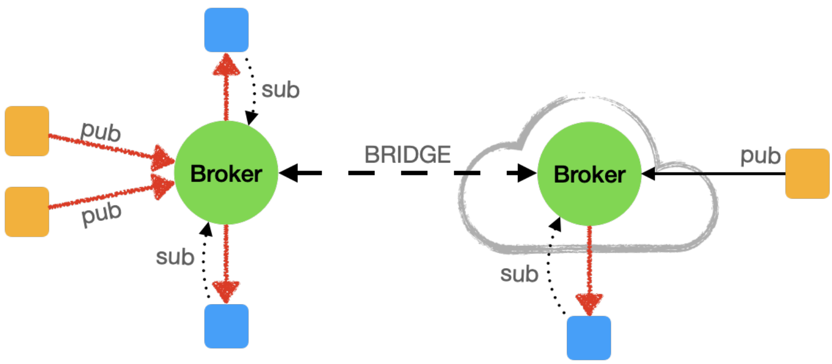



Figure 4.4: An MQTT Brokers bridge from [18].

## 4.5 THE MONITORING SYSTEM FOR LONE WORKERS

The goal of this section is to describe the architecture, the hardware, and the software of the proposed solution, designed to achieve the features stated in the previous sections and summarized as following:

- Verifying that workers effectively wear the helmet, gloves, and protective footwear to prevent accidents, and alerting the safety professionals if these conditions are not met;
- Alerting the rescuers as soon as the sensor detects a human fall or a worker deliberately calls for help, informing rescuers of the victim's position for a fast response.

*The Fog-Computing Architecture*

Figure 4.5 schematizes architecture layers.

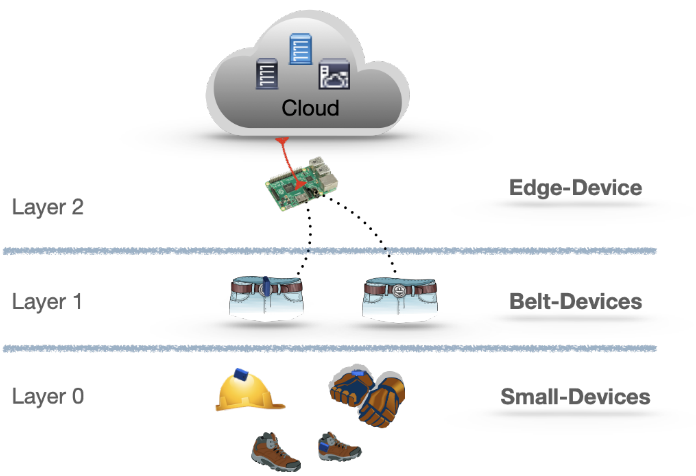



Figure 4.5: The *Fog-Computing* architecture (Source: [18]).

The *Small-Devices*, which make the PPE smart, represent the lower layer and are the end nodes of the architecture.

The *Belt-Devices*, worn by each worker, represent the first layer of *Fog-Nodes*, providing connectivity and processing resources to the lower nodes.

Finally, a single *Edge-Device* connected to the Cloud resources through the Internet connection is the upper layer of *Fog-Nodes*.

*Hardware Components.*

SMALL EMBEDDED DEVICES. Made up of a Bluetooth 5.0 Low energy module, they are attached to each PPE. They form a Personal Area Network (PAN) with the associated *Belt-Device*, as shown in Figure 4.6, depicting a detail of the network.

Figure 4.6: A personal area network from [18].

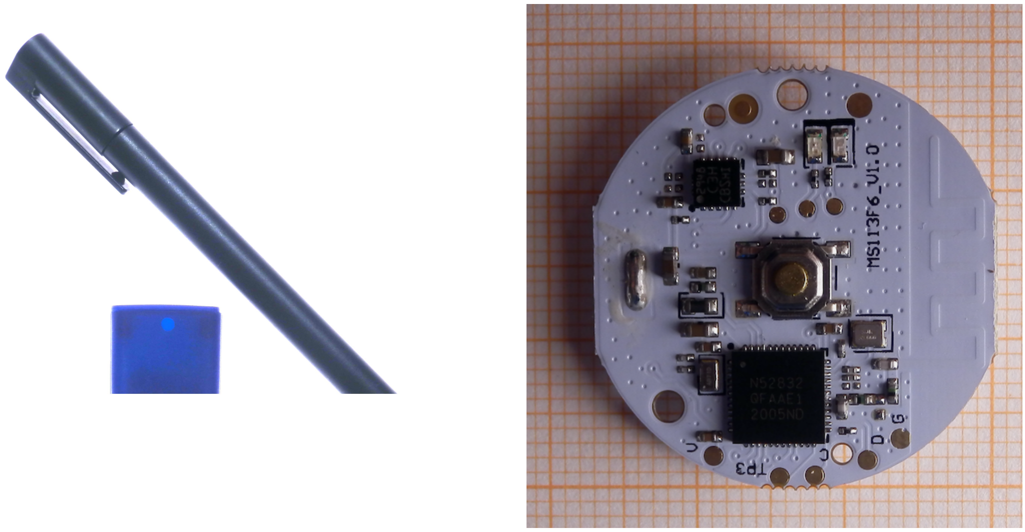
They are not invasive by design, and the fiber of gloves, shoes, and helmets can embed their cases of soft silicon rubber (Figure 4.7.)

These devices, powered by a primary (non-rechargeable) lithium battery, can last years thanks to their low power. The long-lasting battery for these devices is critical in reducing the maintenance and cost of ownership.

BELT-DEVICES. They have an accelerometer, a GPS module, Wi-Fi, and BLE connectivity. Their user interface has one push button, a tricolor LED, and a vibrating device. (Figure 4.8).

A prototypical experimenting system was made up of an ESP32 module [47], featuring a rich Micro Controller Unit (MCU) with integrated Wi-Fi and Bluetooth connectivity, an ADXL345 accelerometer [2] circuit, and an additional GPS module [105].

The size of the *Belt-Devices* (Figure 4.9) is adequate to be used with protecting gloves. They are rugged and silicon-covered to be



(a) Embodiment (b) Printed Circuit Board

Figure 4.7: Small-Device prototype from [18].

resistant to fall and water. Powered by a lithium-ion rechargeable battery, they can last more than one working shift.

EDGE-DEVICE. It offers Wi-Fi connectivity and performs as a local server for Web applications and micro-services. Additionally, it acts as a local MQTT *Broker*, where all locally connected devices publish and subscribe. Whenever one *Edge-Device* cannot guarantee full Wi-Fi coverage in large sites, additional Wi-Fi extenders can expand the covered area.

The *Raspberry Pi* with its *Raspberry Pi OS* is a straightforward and cheap solution to act as the *Edge-Device*. The Raspberry operating system is a Debian Linux distribution compliant with the *Mosquitto* MQTT Broker and all necessary packages to obtain a Linux, Apache, MySQL, and PHP (LAMP) server.

LAMP is a proven set of software to deliver high-performance Web applications. *Mosquitto* is an open-source MQTT broker of Eclipse foundation [81, 135], which also allows the use of *WebSockets* [67]. *WebSockets* are a two-way communication model between a browser and a server over TCP without opening multiple HTTP connections.

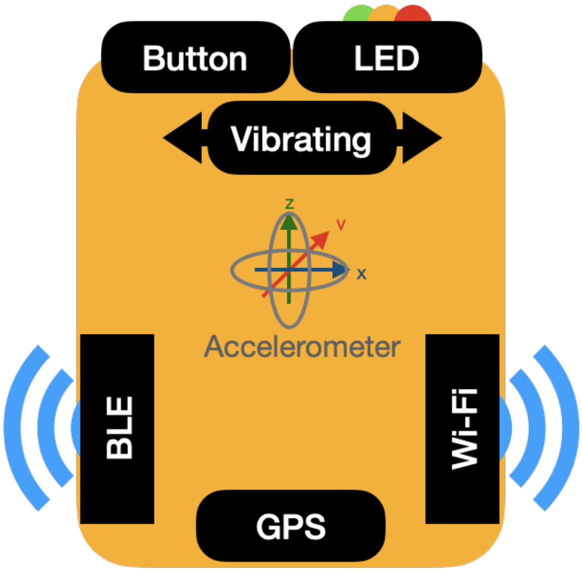


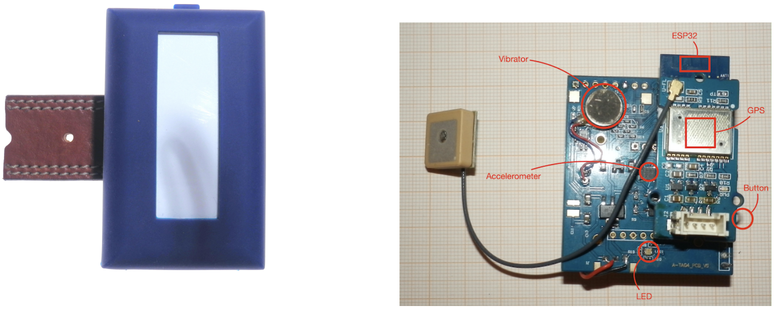

Figure 4.8: A Belt-Device schema from [18].

In [79] the authors suggest a correlation analysis of end-to-end delay and message loss under different mosquitto QoS levels and payloads, allowing an appropriate QoS selection. Thus, since repeated messages and a slight delay do not affect the results, QoS 1 has been chosen.

Furthermore, the *Raspberry* boards have programmable General Purpose Input/Output Ports (GPIO), usable to activate sounding alarms triggered by *Emergency Events*. Finally, an *Access Point* connected to the Ethernet port of *Raspberry* creates a Wi-Fi local network area with a ray ranging from 75 to 100 m in an open place.

### *Networking*

The *Small-Devices* communicate with the associated *Belt-Device* by BLE technologies, creating a PAN for each involved worker. The *Belt-Device* communicates with the *Edge-Device* by Wi-Fi connection and MQTT protocol. The *Edge-Device* communicates with remote and Cloud resources by the Internet connection (Figure 4.10).




(a) Embodiment (b) Printed Circuit Board

Figure 4.9: Belt-Device prototype from [18].

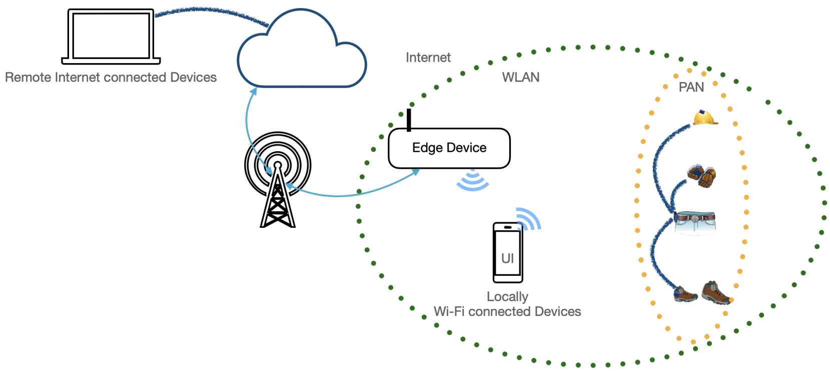



Figure 4.10: The whole network architecture. (Source: [18])

## 4.6 THE SOFTWARE

Communication among the components of the system is the most crucial part of the solution. The devices are stand-alone *Fog-Nodes* that communicate among them delivering their services.

**COMMUNICATION BETWEEN SMALL-DEVICES AND BELT-DEVICE.** It is accomplished by BLE, offering low-cost and interoperable wireless connectivity for compact battery-powered applications. The technology is highly efficient, minimizing the energy required to transfer data. It is secure, specifying features to ensure confidentiality, integrity, and privacy.

Bluetooth technology uses the principles of "inquiry" and "inquiry scan" devices. According to the Bluetooth Core Specification [22], a *BLE Client* application can discover services offered by other *BLE Server* devices in its proximity. The *Small-Devices* embedded in the PPE behave as *BLE Servers*, offering *Alert Notification Services (ANS)* [21] to the devices in their proximity. The *Belt-Devices*, instead, behave as a *BLE Client* that will use those services. Once the *Belt-Device* has discovered the BLE Servers in its proximity, memorizes such connections, alerting the application when one of these connections is weak or lost. Since the *Small-Devices* have a switch that turns the circuits on when the embedding PPE is worn, a loss of connection indicates dismissed safety gear.

COMMUNICATION BETWEEN BELT-DEVICES AND EDGE-DEVICE. As the MQTT communication is bidirectional, *Belt-Devices* can *publish* and *subscribe* at the same time. In this way, the same *Belt-Device* can publish messages to *Edge-Device* but it can also receive messages from other networked services.

### *The firmware*

The firmware makes the *Small-Devices* and the *Belt-Devices* smart devices.

The *Small-Devices* have a firmware that enables the ANS of a BLE server [21]. The *Belt-Device's* firmware, instead, has multiple features, as described in the following. Figure 4.11 shows the flow chart of the *Belt-Device* firmware.

In the beginning, an MQTT client deployed on the *Belt-Device* starts by connecting to the *MQTT Broker*, i.e., the *Edge-Device* of the site, and subscribes to the *Topics* of interest to receive messages from the system.

After this, the flow continues in two parallel paths. The former actuates the device outputs, LED, and vibrator, to produce visual and haptic feedback when a new message arrives from the system. The latter evaluates if publish an alert whenever an accidental human fall, a button press event is recognized, or a BLE connection is lost.

To verify that the PPE is worn all the time, a BLE client deployed on the *Belt Device* associates the initially worn PPE with itself by memorizing the list of BLE servers offering ANS in its proximity.

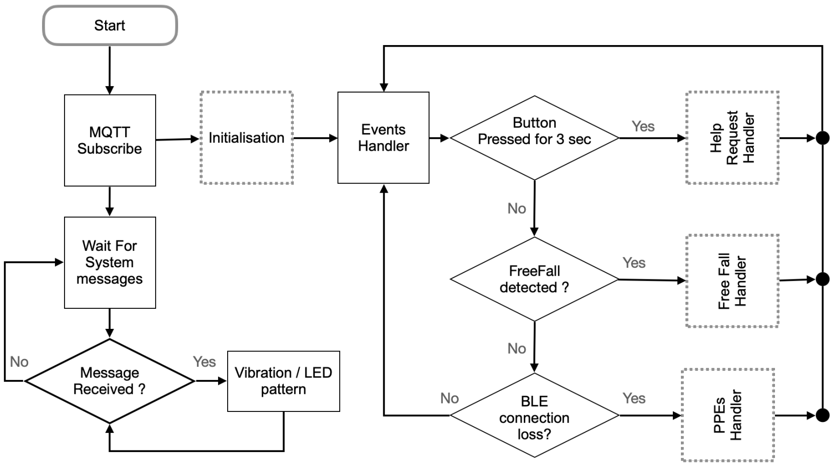



Figure 4.11: Belt-Device firmware flowchart from [18].

Since the *Small-Device* embedded in the PPE offers ANS only if the PPE is worn, when any of the BLE server signals are lost, the *Belt-Device* supposes that the related PPE is not worn or left far from the worker or damaged. In these cases, should be sent a warning message to the *Edge-Device*, but not immediately. Indeed, first, a vibrating output informs the worker that some PPE is not detected. If the worker presses the button of the Belt-Device one time, for three seconds, before ten seconds expire, the message will be delayed for fifteen minutes or canceled if is restored the BLE signal after this period. This strategy allows the worker to take any PPE off for a limited time without warning the supervisor. To allows the worker to dismiss the PPE voluntarily, for example, during lunchtime or at the end of shift, and at the same time take note of event for further analysis, the worker can press the button three times after vibration then, an info message substitutes the warning one. The procedure will restore when the BLE signal is detected again (Figure 4.12.)

The accelerometer chip uses an embedded algorithm to detect human falls. It raises an interrupt when the acceleration on the three axes has a pattern compatible with a human fall [71]. Whenever the Microcontroller Unit (MCU) detects the human fall interrupt, it informs the worker by generating a long vibration. The *Belt-Device* publishes an alarm message containing the GPS coordinates to notify the rescuers of the event and place if the

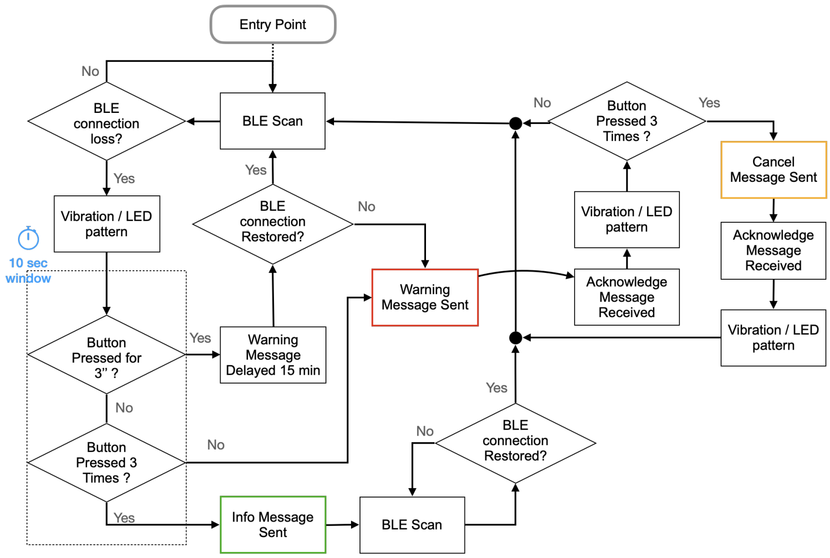



Figure 4.12: PPE worn check, flowchart from [18].

worker does not cancel the event by keeping the user button pressed for three seconds within 10 s from the haptic feedback (Figure 4.13.)

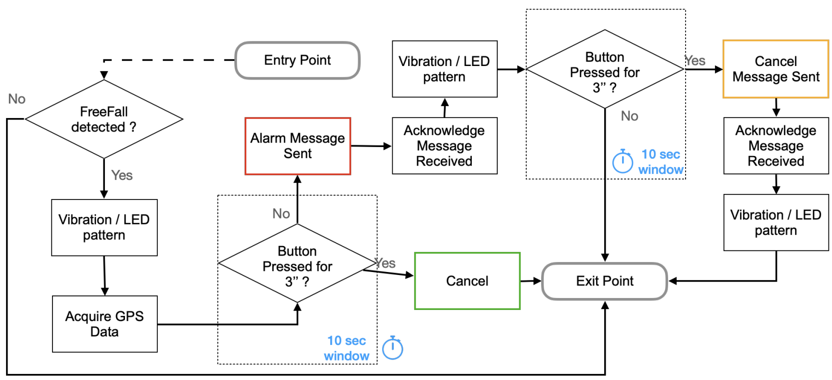



Figure 4.13: Human fall check from [18].

The worker always has the opportunity to cancel the alarm before sent or soon after.

The firmware allows interaction with the user, managing the push button, the vibrating device, and the tricolor LED.

The user intervenes to:

- call for help and rescue, pressing the button of the *Belt-Device*. When the *Button Press Event* raises, the firmware verifies that the user keeps the button pressed for at least three seconds before publishing an Emergency message to the *Edge-Device*. This precaution prevents the raising of alarms due to involuntary temporary button press.

- To cancel a false alarm, the worker, after receiving from the *Edge-Device* the acknowledgment of previously published alarm message, can press the button of the *Belt-Device*. The acknowledgment is a vibration of the *Belt-Device* three seconds long. If the worker presses the *Belt-Device* button within ten seconds, a new message from the *Belt-Device* to the *Edge-Device* will abort the Emergency condition (Figure 4.14).

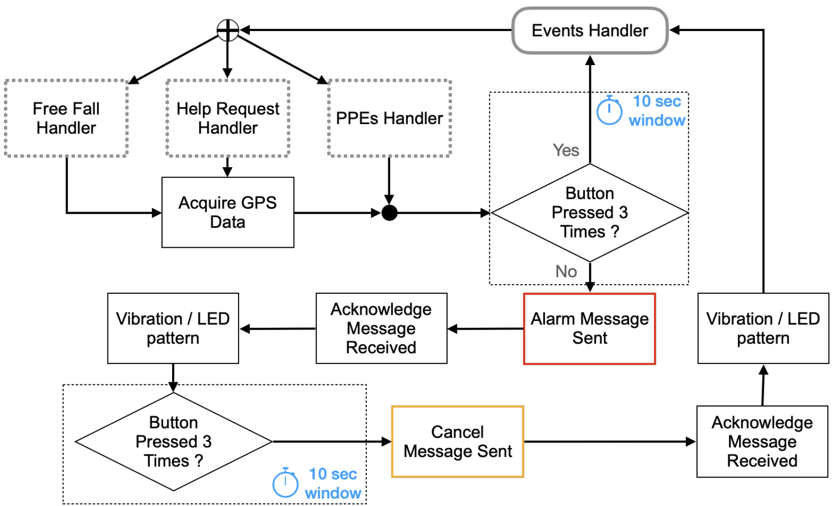



Figure 4.14: False alarm cancellation flowchart from [18].

After a human fall or a help request, the *Belt-Device* GPS module outputs the worker's position, with an National Marine Electronics Association (NMEA) standard string [46]. The NMEA standard is a set of formatted lines, named sentences, easily readable by the MCU serial port. As an example, a sentence can contain the following information:

- Time: 235317.000 is 23:53 and 17.000 s in Greenwich Mean Time.

- Longitude: 4003.9040,N is latitude in degrees.decimal minutes, North.

- Latitude: 10512.5792,W is longitude in degrees.decimal minutes, West.

- Number of satellites seen: 08.

- Altitude: 1577 m.

By processing it, the MCU can extract *Time*, *Longitude*, *Latitude*, and *Altitude*, and adds them to the *Emergency message*. It is worth noting that the workers' position is communicated to the *Edge-Device* only in these cases, preserving their privacy otherwise.

Although the rescue time depends on many external factors, such as the distance of rescuers from the place of accident or the difficulties in reaching the accident place due to factors such as weather or terrain conditions, a prompt alert of accident occurrence, and precise knowledge of the location to run, undoubtedly accelerate the rescue. This faster reaction can result in life-saving. For example, a *Suspension Trauma* resulting from a human fall can occur in as little as 20-30 min, and sometimes less, depending on a person's health and the nature of injuries sustained in the fall. The rescue must begin immediately, as also stated in [123].

*The Web application.*

The Web application can visualize the workers' status and possible real-time alarms. Any computer device with a modern browser can access a web dashboard and get the workers' real-time status. Figure 4.15 shows a schematic view of the whole system and how its software applications communicate.

The web interface uses a multi coordinate view paradigm, summarizing the state of each worker as a simple list. Whenever it becomes critical, it is geo-localized and synchronized on a map, as shown in Figure 4.16.

The connection between the mobile devices and the Apache2 web server uses the Secure Sockets Layer (SSL) protocol to guarantee the security and privacy of exchanged information, even if on a local area network. Thus, the Web application utilizing the Hypertext Transfer Protocol Secure (HTTPS) protocol offers native

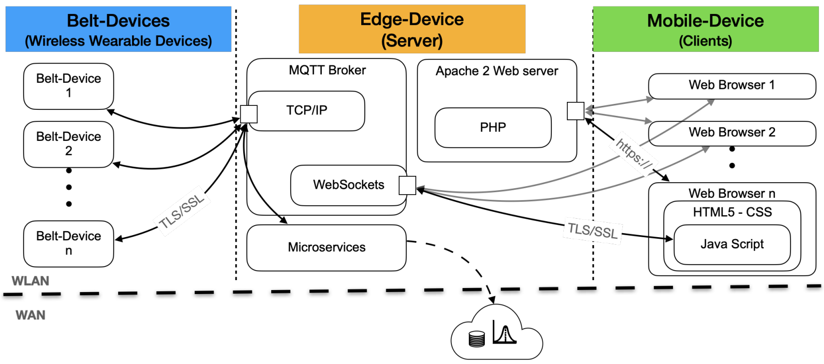



Figure 4.15: System software overview (Source: [5].)

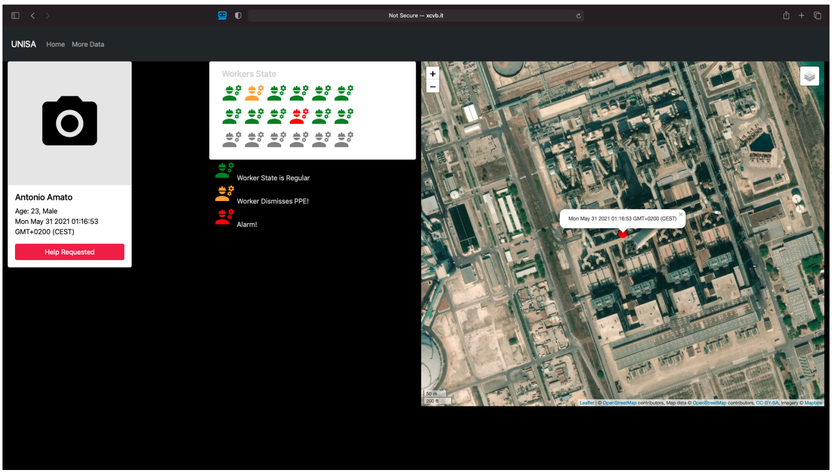



Figure 4.16: The Web application screenshot (Source: [5].)

privacy protection for all the communications between mobile devices and the *Edge-Device*.

For the communications between the wearable *Belt-Devices* and *Edge-Device*, the solution leverages the MQTT protocol's security features, offering resolutions on three levels:

- At the Network level, the data exchanged between *Belt-Devices* and *Edge-Device* happens on the local Wi-Fi area network. Thus, all the Wi-Fi security protocols prevent the *man in the middle* attack. The connection with the Cloud resource, if applied, can occur through a Virtual Private Network (VPN) connection for additional security, if required.

- At the Transport level, Transport Layer Security (TLS)/SSL is used for encryption. This method ensures that data cannot be read during transmission and provides client certificate authentication to verify the identity of both parties.
- At the Application Level, the MQTT protocol provides client identification and username/password credentials to authenticate devices.

*Microservices.*

They realize business logic and back-end operations. Each MQTT message published is an event. Thus, some events activate some actions while some other events need only to be displayed or notified.

Since the *Microservices* can publish and subscribe to the *Topics* on the same *MQTT Broker* used by the Web application and the *Belt-Devices*, they can analyze all the published messages and react accordingly. In this way, the MQTT broker directly manages the events that need to be visualized or notified to the supervisors, pushing them to the Web page on the connected browsers through websockets. Instead, the *Microservices* manage the events that require actions. They publish an acknowledgment for each received event as *Rescue request*, *Human fall revealed* (Emergency messages type), *PPE dismissed* (Alert messages type), Alarm canceled. Furthermore, the *Microservices* can activate an output signal on the *Edge-Device* GPIO to actuate an audio alarm if one of the Emergency message types is published. Optionally, the Microservices can persistently log all the events onto Cloud storage for further analysis.

## 4.7 RUNNING THE MONITORING SYSTEM

This section briefly illustrates the proposed solution deployed at an outdoor construction site.

Deploying the *Edge-Device* at the worksite office, the power supply for it is available easily. Then, one (or more) access point creates a wireless local area network (WLAN) covering all the site areas. If only one access point is enough, can stay in the office. If the Wi-Fi extension needs to cover a larger area, the extenders can be powered by a battery and solar panel and placed on movable poles. However, the power is usually available around the site for illumination requirements or electric-powered tools, so the Wi-Fi extenders could also be powered by those sources.

When a team of workers reaches the construction site, they put their smart PPE on and wear any of the charged *Belt-Devices* taken from the charging dock stations.

Before starting to work, the worker initializes the *Belt-Device* by pressing its button. The initialization consists of the discovery phase, performed by the *Belt-Device*. In this phase, the *Belt-Device* looks for the closest BLE servers, offering ANS, and registers these servers as the PPE to monitor. During this phase, the workers must keep a given distance among them to avoid that the *Belt-Device* of one worker reveals the PPE of another worker and erroneously identifies it as its one. Starting the initialization phase, while workers cross a turnstile or a gate one by one, could mitigate this problem.

As previously described, when messages reach the *Edge-Device*, they can be read by any client connected to the Web server, and a browser can easily monitor the worker's state. Whenever an *Emergency type* message raises, the *Microservices* on the *Edge-Device* will push a notification to the connected browsers and activate a loud audio alarm to alert the personnel.

Bridging the local MQTT Broker with another remote MQTT Broker on the Cloud allows all the same features remotely.

## 4.8 IN-LAB TEST

The following describes the tests performed to evaluate the network architecture, a critical component of the whole solution.

The tests results show that the *Fog-Computing architecture* offers a wide margin of tolerance to the network capacity. During stressful tests, the system managed up to ≈9 alert messages per second, a capacity much higher than the expected one into an actual deployment. No tests were performed on the Internet connection since it is not essential due to the proposed *Fog-Computing Architecture*, and the entire solution is tolerant by design to the unreliable Internet connection. The proposed hardware design transforms classic PPEs such as helmets, protective gloves, and shoes into smart PPE.

The solution does not require any particular infrastructure. Just the Wi-Fi coverage of the working site is sufficient.

The choice of the MQTT protocol for messaging has proven ideal for connecting low-cost wireless devices with a small code footprint and minimal network bandwidth.

To monitor workers' status and manage the system, a responsive Web application at the front-end allows the use of smartphones, tablets, and notebooks in the field without installing additional software.

Designing a low-cost solution was one of the most critical factors to make it acceptable to employers. Thus, the satisfactory cost/benefit relation is due to:

- affordable costs of the electronics parts of *smart PPE*, and the negligible cost of the *Edge-Device*, the consequence of the electronic components selected among those readily available on the shelf;

- low installation costs, which are limited to some additional Wi-Fi expander when installation is for the extensive work sites;

- the low recurring costs for maintenance and ownership. They are limited to the battery changes of the *Small-Devices*, once every few years.

Since the network plays a relevant role, the capability of the designed distributed architecture has been under test to validate its adequacy concerning the intended purposes, reporting at the end of this section the evaluation of the resulting message exchange capacity against the needs of the proposed solution.

*Materials*

The system under test was made up of the following hardware:

- One *Raspberry Pi 3 Model B* as *Edge-Device* (Figure 4.17).

- One Access Point (AP) TP-Link TL-MR3040.

- One *Belt-Device* prototype.

- Five *Small-Devices* to embed on two PPE.

During the test setup, the *Edge-Device* had the fixed IP address 10.10.1.215, which represented the IP address of the Web application as well the address of the local MQTT broker on the LAN. The AP had the Service Set Identifier (SSID) assigned and credentials to access the WLAN through a Wi-Fi Protected Access II (WPA2) authentication process.

On the *Edge-Device* the so-called LAMP configuration was installed, consisting of the last stable version of an Apache2 webserver with PHP module and the MySQL database management system.

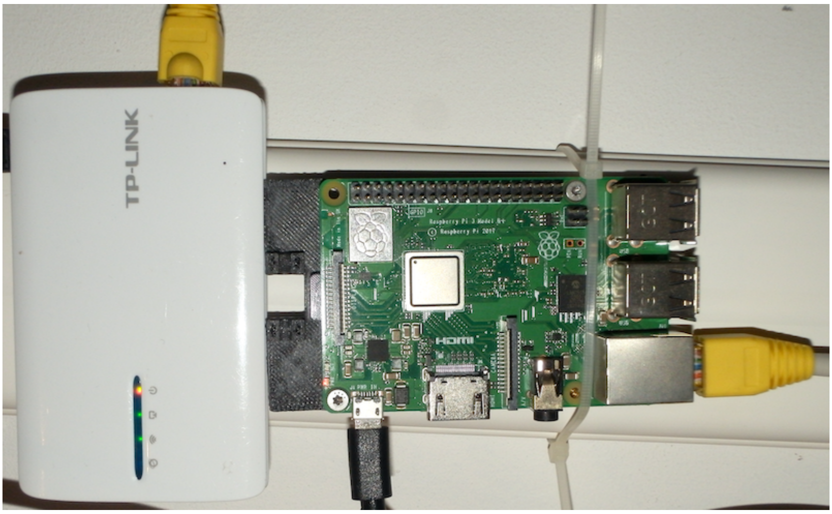



Figure 4.17: *Edge-Device* configuration from [18].

A Mosquitto MQTT broker was installed on the *Edge-Device*, configured with two listeners, one on the default port with MQTT protocol and another on a different port with websockets protocol.

The *Belt-Device* application, was developed by using the *Lua* computer language. Lua is a lightweight, embeddable scripting language. Since the micro-controller of the *Belt-Device* was an ESP32 SoC module, the environment was deployed by using the *NodeMCU* [94] firmware, an open-source *eLua* based firmware layered on the Espressif® NON-OS SDK [42]. To edit and load the application on the *Belt-Device* the *ESPlorer* was used as the Integrated Development Environment (IDE) for ESP32 developers [43].

The features of the *Small-Device* described in Section 4.5 were developed by using the Nordic® SoftDevices for communication and the nRF52 SDK13.0 for development. SoftDevices are pre-compiled, event-driven protocol stacks with APIs that allow the application to interact with the BLE protocol. Nordic® developed, tested, and maintains SoftDevices.

*Methods to Evaluate the Communication Capacity*

Defined the capacity of communication as the number of messages that through the network it is possible to display on a web page in one second, the test measured the capacity to show on the supervisor web page the messages published by the *Belt-Devices*.

A browser on a Personal Computer (PC) connected to the WLAN loaded the test Web page. Once the page loaded, the JavaScript code published the PC current clock time to a specific MQTT *Topic*. The *Belt-Device* received the message and fixed its internal real-time clock. This operation synchronized the two clocks in almost real-time, within a precision of many orders smaller than one second, which is the resolution required for the tests. Since the JavaScript code of the web page subscribed to the topic '*test*' with a web sockets connection, on the web page appeared all the messages published on this topic.

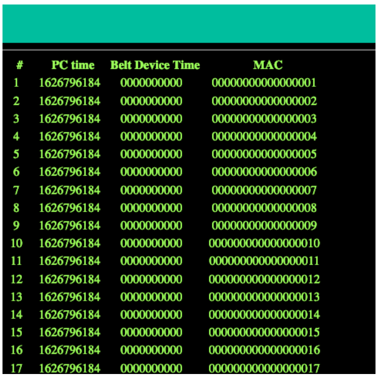
Initially, a simple loop on *Belt-Device* published 2000 messages, which were displayed on the web page, to measure the overall capacity of *Belt-Device* in the publication of messages. The measurements showed that the page was able to display 182 messages per second ( Figure 4.18a.)

This capacity, defined as the *channel capacity*, is the estimation of the maximum quantity of messages per second that the system architecture eventually can display to the supervisors of lone workers.

Since the longer network path involves the *Small-Device*, the *Belt-Device*, the *Edge-Device* up to the browser of the supervisor, was performed the following tests. To evaluate the publishing capacity of the *Belt-Device* while detecting the *Small-Devices* in the proximity, one *Small-Device* was turned on, and the *Belt-Device* forwarded its MAC address to *MQTT Broker* as soon as it detects the *Small-Device* BLE advertise (ADV). The *Small-Device* was set to transmit an ADV each 100 ms, while the *Belt-Device* BLE discovery cycle was set to wait for the ADV for 12.5 ms each 25 ms.

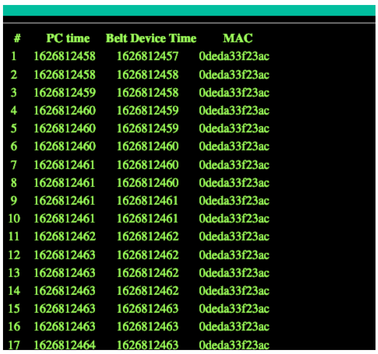
The measurements showed that the page displayed 100 received messages in 33 seconds, i.e., ≈3 messages each second.

Given $D = t_2 - t_1$ as a delay where $t_1$ are the seconds elapsed from the Unix epoch (00:00:00 UTC on 1 January 1970) to the *Belt-Device* ADV detection, and $t_2$ are the seconds elapsed from the Unix epoch to the message visualization on the web page, out of 100 received messages, 61 of them showed a delay $D$ of 1 second, while the remaining 39 messages showed a delay $D = 0$, thus giving an average delay of $0.61 sec$ ( Figure 4.18b).



| # | PC time | Belt Device Time | MAC |
|---|---|---|---|
| 1 | 1626796184 | 0000000000 | 00000000000000001 |
| 2 | 1626796184 | 0000000000 | 00000000000000002 |
| 3 | 1626796184 | 0000000000 | 00000000000000003 |
| 4 | 1626796184 | 0000000000 | 00000000000000004 |
| 5 | 1626796184 | 0000000000 | 00000000000000005 |
| 6 | 1626796184 | 0000000000 | 00000000000000006 |
| 7 | 1626796184 | 0000000000 | 00000000000000007 |
| 8 | 1626796184 | 0000000000 | 00000000000000008 |
| 9 | 1626796184 | 0000000000 | 00000000000000009 |
| 10 | 1626796184 | 0000000000 | 000000000000000010 |
| 11 | 1626796184 | 0000000000 | 000000000000000011 |
| 12 | 1626796184 | 0000000000 | 000000000000000012 |
| 13 | 1626796184 | 0000000000 | 000000000000000013 |
| 14 | 1626796184 | 0000000000 | 000000000000000014 |
| 15 | 1626796184 | 0000000000 | 000000000000000015 |
| 16 | 1626796184 | 0000000000 | 000000000000000016 |
| 17 | 1626796184 | 0000000000 | 000000000000000017 |

(a) Web page from Test 1



| # | PC time | Belt Device Time | MAC |
|---|---|---|---|
| 1 | 1626812458 | 1626812457 | 0deda33f23ac |
| 2 | 1626812458 | 1626812458 | 0deda33f23ac |
| 3 | 1626812459 | 1626812458 | 0deda33f23ac |
| 4 | 1626812460 | 1626812459 | 0deda33f23ac |
| 5 | 1626812460 | 1626812459 | 0deda33f23ac |
| 6 | 1626812460 | 1626812460 | 0deda33f23ac |
| 7 | 1626812461 | 1626812460 | 0deda33f23ac |
| 8 | 1626812461 | 1626812460 | 0deda33f23ac |
| 9 | 1626812461 | 1626812461 | 0deda33f23ac |
| 10 | 1626812461 | 1626812461 | 0deda33f23ac |
| 11 | 1626812462 | 1626812462 | 0deda33f23ac |
| 12 | 1626812463 | 1626812462 | 0deda33f23ac |
| 13 | 1626812463 | 1626812462 | 0deda33f23ac |
| 14 | 1626812463 | 1626812462 | 0deda33f23ac |
| 15 | 1626812463 | 1626812463 | 0deda33f23ac |
| 16 | 1626812463 | 1626812463 | 0deda33f23ac |
| 17 | 1626812464 | 1626812463 | 0deda33f23ac |

(b) Web page from Test 2

Figure 4.18: Screenshot of Web pages during the tests (source: [18].)

*Tests Results*

Any of the sent messages were lost, and the timing was stable as reported, even repeating several times the tests. Since the occurrence of events like smart PPE dismission, accidental worker falls, or worker calls should be few, almost very rare, the *Belt-Device* will not publish a critical quantity of messages. Indeed, the measured communication capacity of the whole system is many orders higher, even considering some connection problem that would oblige the *Belt-Device* to resend the message multiple times. Likewise, in the unlikely case of a large number of events simultaneously, for example, 100 lone workers requiring rescue concurrently, the tests showed that the *channel capacity* is still more than enough.

Furthermore, it is worth noting that even a delay of one second from the event and the reception of its message on the supervising system would not invalidate the advantage of the proposed solution. Repetition of previous tests with five *Small-Devices* detected by the same *Belt-Device* validates the hypothesis that a more significant number of *Small-Devices* in the proximity of one *Belt-Device* not exceed the channel capacity.

### *Test Conclusions*

The measurements showed that the page was able to display 100 messages in 11s, i.e., ≈9 received messages each second. Considering that only multiples of seconds were measured, out of 100 received messages, 85 of them had a delay $D = 1$s, and the remaining 15 messages had a delay of $D = 0$, giving an average delay of ≈0.86s. The following graph, Figure 4.19, shows the distribution of messages among the five *Small-Devices*, displayed by the web page in 11s.

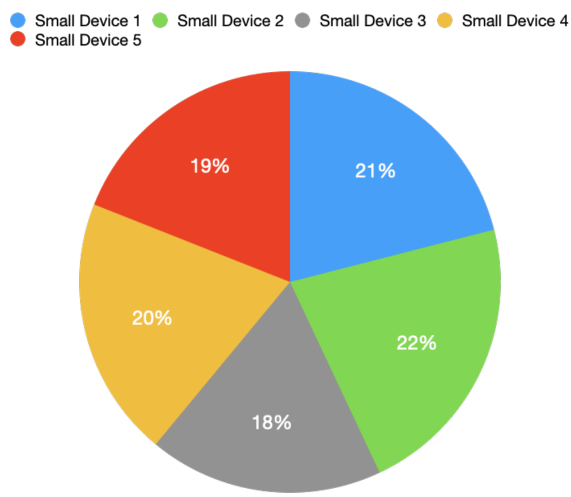



Figure 4.19: Messages distribution among the *Small-Devices* (Source: [18].)

Even with five *Small-Devices* simultaneously monitored, the *Belt-Device* can communicate their connections to the Edge-Device, continuously and almost uniformly, occupying still a small part of the channel capacity.

## 4.9 CONTROL ROOMS COOPERATION

At AVI 2020 Conference, we presented the project described in [9]. Its main goal was to demonstrate *Karya Advanced Sensor*, an IT solution to locate firefighters during operations in a forest fire. *Karya Advanced Sensor* is a networked and wearable Smart Badge conceived to safeguard the firefighters when they operate isolated and close to the flames, seriously endangering their safety [108]. The work proposed the use of the MQTT protocol for real-time messaging among all operators engaged in the operations. The operators stay on the field and at a remote Control Rooms supervising and coordinating all the procedures .

*Karya Advanced Sensor* is a system provviding safety services to fire-fighting operators and to automaticaly deline the areas covered by the fire. It is part of a digital ecosystem to facilitate proper management of the forest emergency and to improve the safety of firefighting operators. In particular, the smart badge is equipped with a heat detector, fall sensor, and GPS module integrated with technical vest of the staff. The system, based on a Mist-Fog Computing architecture, is depicted in Figure 4.20.

When the firefighter works in an area where the temperatures are very high, the gateway, present on the fire department tanker, detects the badge data via a Wi-Fi network. The operator on alert near the tanker will have full vision, through his tablet, of the situation and the movements of his colleagues, engaged in extinguishing the fire, up to a distance of 100 meters, this distance, considered safe for operators. The team in the field, via a 2.5 G Global System for Mobile Communications (GSM) network, will send the data and information first to a cloud system, which in addition to keeping the data permanently for subsequent processing, will make the information available in the control rooms. So, if an operator's sensor detects a fall the system sends an alarm to the personnel on board the truck and to the system on a Multitouch wall in the control rooms, providing

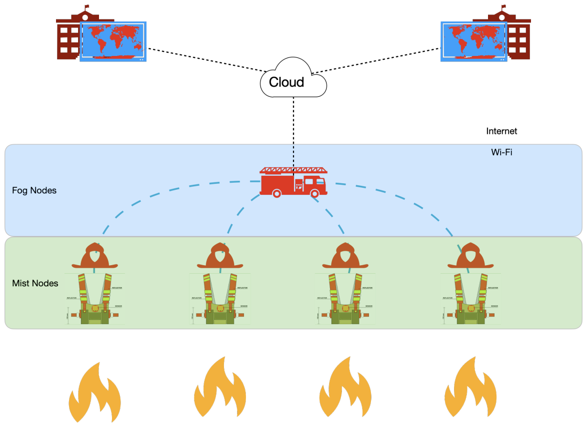



Figure 4.20: *Karya Advanced Sensor*: the system Mist-Fog Computing architecture

information (e.g. GPS position and cause of the fall) useful to reach the person with the appropriate care quickly. Besides, the badge can define an area covered by fire. The sensor will save the trace of workers' movements if the temperature exceeds the set threshold; the georeferenced tracks will define the movements of the firefighting operators. Fire-fighting operators' movements are traced back whenever the temperature exceeds 50°C. Furthermore, the threshold of the heat detector can be set according to the latitude of its use. Usually, the activation, and therefore the saving of the traces, takes place after the threshold of 50 ° C, this can be increased if the sensor is used in warm areas of the world, or vice versa lowered in colder areas. Finally, their positions are sent to the system on the Multitouch Wall in the control room and processed by an algorithm to define a polygon, corresponding to the perimeter of the forest fire. Through the interface on Multitouch Wall, a thematic map is available, positioned on an area of interest, which provides the operators of the control room with information about the orography, water sources, housing settlements, obstacles above ground, and weather forecasts. In addition, the data sent by the teams of the firefighters on the territory of the emergency, are combined in order to allow an analysis of the data, useful to the employee who can thus not only intervene promptly but also prevent problems. The data are displayed as visual indicators of immediate interpretation to allow an intuitive approach to the system. The immediate interaction, the ability to customize the view of the area, and the wealth of contents may greatly facilitate the decisions of the operators.

Although this research focused on the safety of the specific lone worker and investigated advanced sensor and location techniques, here is mentioned because it represents the base for a multi-application coordinated view, particularly suitable for control rooms but not only. The events and messages of the *Karya Advanced Sensor* are communicated through MQTT protocol, and the research described in the following Chapter 7 leverages this experience.

Part II

# GEOVISUAL ANALYSIS

# INTRODUCTION

The first part of this work investigated and experimented with the emerging architecture to leverage the pervasiveness of IP-connected devices since the massive quantity of data they produce is a precious source of information about people's status and the environment where they live.

Most of the data encountered in these experiments were spatiotemporal and produced at different rates, ranging from many data per second to a few data per day. Using the technologies reported in this part of the dissertation, it was possible to temporally decouple producers from consumers so that data was asynchronously available.

With tools, such as Apache Hadoop, Spark, and Kafka, it was possible to create a distributed and scalable architecture. It was possible to analyze data from streaming and databases, visualizing them with web technologies even in real-time.

Building on this experience, the second part of the thesis studies technologies to visually analyze this data, with the idea that visual tools are easier to understand and use than commands or menu-based solutions. A simpler solution should facilitate access to information by non-expert stakeholders and citizens. To do so, in the next Chapter 5 has been studied and experimented systematic user co-creation approach generating research and innovation processes in real-life communities and settings; the living labs. Then in Chapter 6 are proposed *Chorems*, to visually synthesize dynamic phenomena, introducing a use case. Finally, in Chapter 7 the new proposal to support easy and intuitive geovisual analysis of the territory is described.

# 5 AN OPEN INNOVATION PLATFORM TO SUPPORT LOCAL DEVELOPMENT POLICIES

## 5.1 SUMMARY

This chapter presents the work done to experiment the use of Living Labs (LLs). A Living Lab (LL) is usually a platform for research that is, however, based on a specific geographic location or territory. Such a place then becomes the testing ground for innovation and co-production of knowledge on sustainability. Therefore, web platforms to inform and involve all stakeholders during decision-making processes or the development of territorial solutions become essential to facilitate the citizens' participation. The chapter describes the realization of a platform with tools that allow collecting data automatically from National Institute of Statistics (ISTAT) and to perform some typical urban planning analysis on them.

## 5.2 INTRODUCTION

A LL represents a systematic user co-creation approach that integrates research and innovation processes in real-life communities and settings. This chapter focuses on the role that LLs can play in strengthening the skills of territory by increasing knowledge deriving from it and improving users' engagement for sustainable innovation, as published in [7].

An innovative approach to the design and use of Living Labs is studied. It aims to integrate territorial intelligence and Information with Communication Technologies (ICT), to better understand territorial phenomena and interpret local dynamics involving citizens, institutions, and organizations.

Follow a description of a technological solution conceived to increase the automation process for knowledge creation and sharing. It allows users to query thematic datasets and visualize (geo-located) information. Each information produced by aggregating and mining procedures, in turn, can be integrated as a resource belonging to the heritage of the spatially enabled territory.

IT enables citizens to contribute to a new form of participatory democracy where private actors, associations, and individuals participate in decisions-making and bring together their experiences to enrich local resources. This form of citizen engagement is currently known as collective intelligence and represents an indispensable step toward actualizing the three pillars of the Open Government (OG) paradigm, namely transparency, participation, and collaboration [26, 92, 98, 132].

Examples of how people are involved in these community-based activities are now common enough, and they mainly refer to the civic consultations in defense of cultural and environmental heritages. However, the scope where the collective intelligence may be effective ranges from the natural environment to healthcare, from safety to mobility, thanks to the pervasive impact of the unifying element of those domains, namely the territory underlying a community and the information deriving from it. The geographic information plays a relevant role within the collective intelligence building process since it synthesizes and aggregates homogeneous data sets, preparing a base to enable people to cooperate for a problem solution. When such shared collections of data heritage generated by territory are then regularly organized through a system of models, methods, processes, people, and tools, the collective intelligence assumes a specific territorial connotation and becomes Territorial Intelligence (TI) [75, 110, 111].

The term TI relates to the development of territory and communities acting on it to pursue its valorization through a continuous improvement of its economic, social, environmental, and cultural (sustainability) dimensions. This claim explains the essential role that TI is gaining for organizations and companies belonging to the same geographic area. Indeed it works as a collector of information deriving from multiple sources and, by processing it, TI produces knowledge relevant to the underlying territory and supports the exchange of know-how among local actors of different cultures [20]. Moreover, activating the citizens' participation in the civil life of the territory modifies their role from only information consumers to information prosumers, i.e., producers and consumers at the same time. It is known in fact that citizen's contribution to collect meaningful data about an event has turned out to be significant in many situations, such as the volunteered geographic information provided by the Haitians after the devastating earthquake in 2010 that notably contributed to the (digital) humanitarian community [124].

Managing the multidisciplinary complexity of data coming from a spatially enabled territory and supporting a profitable exchange of data are the objectives of the work described in this chapter. It is to realize services by integrating four primary territory-oriented elements, namely content, communities, practices and policy, and technology, by investigating a digital ecosystem as an open and shared environment with properties of scalability and sustainability.

An innovative approach to the design and use of LLs is studied to understand territorial phenomena and interpret local dynamics involving citizens, institutions, and organizations. When properly collected, shared, and crossed with data on users and services through IT tools, TI can provide decision-makers with appropriate territorial indicators helpful to conceive new development models aligned with sustainable development.

## 5.3 LIVING LABS

When faced with complex challenges in an evolving real-life context with multiple factors to consider, it becomes hard, if not impossible, for a single actor to find the right solution. For this reason, the LLs strategy becomes the best choice to adopt as it offers the possibility to design solutions with end-users and stakeholders through collaboration and co-creation. In this way, the chance of finding a sustainable and advantageous solution increases, reducing the complexity and uncertainty.

MIT's Prof William (Bill) Mitchell used the LL term for the first time. It originates from purpose to built labs where volunteer research participants were provided with all facilities of a regular home and were observed with all sorts of devices while testing new technologies [50]. The concept evolved over the years, and currently, it bases on a systematic user co-creation approach that integrates research and innovation processes in real-life communities and settings. Instead of a laboratory setting, observation of users takes place in their everyday environment. This "living" approach, placing the citizen at the center of innovation, improves the ability to benefit from the opportunities offered by new IT solutions to satisfy requirements coming from local contexts.

According to the European Network of LLs [50], 5 key elements are paramount for a LL, namely

1. active user involvement,
2. real-life setting,
3. multi-stakeholder participation,
4. a multi-method approach, and
5. co-creation

Elements 3 and 4 emphasize the creation of cross-fertilization, combining methods and tools originating from several and diverse disciplines, involving technology and service providers, and institutional and private actors.

ENoLL is the international federation of benchmarked Living Labs in Europe and worldwide, founded in November 2006 under the auspices of the Finnish European Presidency. The European approach to LLs was created in the Unit "Collaborative working environments" of the DG CONNECT updating, the original LL concept to open innovation environments to attract intellectual and financial investments [50]. Today, ENoLL counts over 150 active LLs members worldwide and is present on five continents. It provides co-creation, user engagement, test and experimentation facilities addressed to innovation in different domains, such as energy, media, mobility, healthcare, and agrifood. Figure 5.1 shows the distribution of LL members in 2015. The most common topics are Health and Wellness, Social inclusiveness, Smart Cities, and Social Innovation. Other themes of interest are Education, Mobility, Energy, Public administration, Culture, and creativeness.

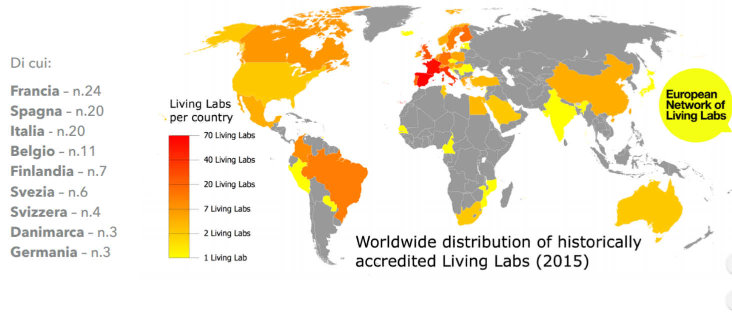



Figure 5.1: The 2015 world distribution of LL members (Source: [50])

*The LL architecture, resources, actors and roles*

A LL is a distributed, adaptive, open socio-technical system with self-organization and sustainability properties. It embeds a set of satellite applications, Web portals, external systems, legacy systems, and corporate Data Base (DB)s, representing the Technology and Infrastructure component that integrates the LL and the IT Infrastructure and Hardware meta-elements.

Grouping users according to their role facilitates the management of the services assigned to them. Single users, thematic communities, organizations/institutions, and regulatory bodies are generic examples of clusters of users who access and produce information through the LL component that coordinates the aspects related to fruition, competition, and collaboration for resources among diverse entities.

Figure 5.2 depicts the general schema of an "ideal" LL showing the interactions between the various hardware, software, and "human" resources.

Data already extracted, standardized, cleaned, and loaded in a database are associated at the lowest level, ready to be saved, managed, and queried through drivers made available by various programming languages.

A mediated schema between the various databases and servers is necessary at the second level. It is created through a data integration process presenting the illusion of having a single endpoint for accessing information. The task of this layer is to receive the queries from the servers, send them to the individual databases involved, collect the results in a single response, and

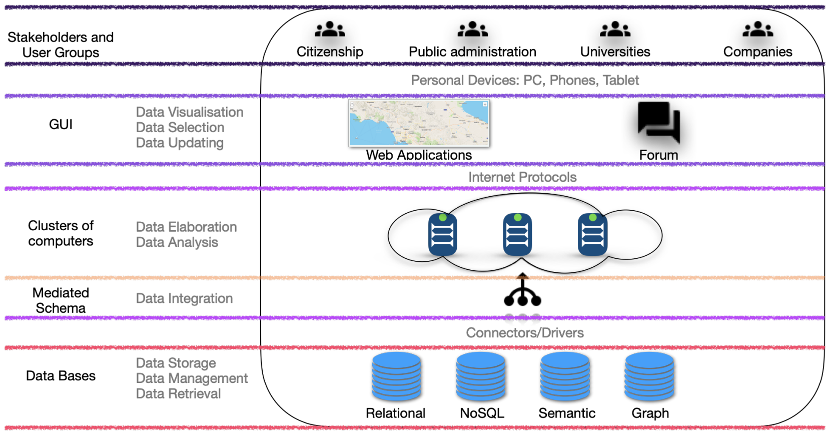



Figure 5.2: A general schema showing a LL architecture and its interactions (Source:[7])

send them back to the servers. The layer plays a fundamental role in gathering requests sent by the user, which may concern the simple retrieval of already available data or new knowledge production through data analysis processes. The transmission of requests and their answers occurs through the classic transmission protocols adopted for the web.

The Graphic User Interface (GUI) has a Geographic Information System on Web (Web-GIS) component to support the research and development team's activities, and an additional users-forum solution. The latter connects all community members through easily accessible devices such as PCs and smartphones. The application visualizes traditional data in tables and spreadsheets and georeferenced data on maps. It also allows the basic operations of insertion, modification, and deletion. Moreover, it allows users to apply clustering algorithms, statistics functions, graphs creation, and other data operations.

Other required features are:

- a "Registration" section to enroll each person;
- a "Sharing" section where share news, files, objectives, and results;
- a "Learning" section allowing users to attend courses;

- a "Talking" section where it is possible to discuss on a particular topic or problem, expose ideas, opinions, and knowledge, answer direct questions, propose surveys and questionnaires that can be completed by all or only a group of members.

## 5.4 FROM DATA TO KNOWLEDGE: DATA SUPPLY CHAIN FOR LLS

The primary component that gives life to a LL is undoubtedly the "Data," consisting of the contribution of all people participating actively and passively. This Section focuses on methods and tools conceived to collect, standardize and clean data, and be fruitful support to the community cooperation.

Billions of people worldwide use the Internet for the first time on a mobile device. Therefore, delivering native-app quality in Web applications that are reliable, fast, and engaging is an essential tool for a LL.

A general framework valid for any application domain is proposed in Figure 5.3, describing all the steps required to build a queryable knowledge base capable of supporting research and development activities.

The first and more challenging step is data collection. Each person in a LL community can contribute to putting data and transferring their personal experience.

The acquired data could be already clean and communicated in a standard way, such as data coming from Open Data sources by governments, or need Extract, Transform, Load (ETL) phases [38]. Even from other databases and Web API is possible to acquire structured data. Alternatively, during the data *Extraction* phase, it is possible to acquire data from unstructured sources, such as text documents, Web pages [99], messages, and more. Data extracted from the source is raw and needs to be cleansed, mapped, and transformed into a defined standard structure. The *Data Cleaning* process is invoked during the *Transformation* phase. It is addressed to clean data from redundancy and inconsistencies and represents a sophisticated process that can be automatically performed, as proposed in [101], and supervised.

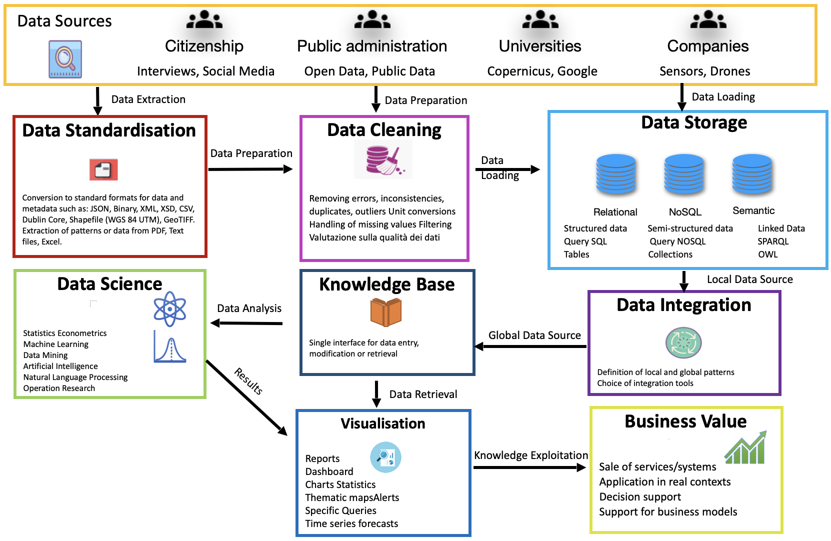



Figure 5.3: Integration and analysis of data from multiple sources (Source: [7])

The *Cleaning* process addresses the management of missing or partial values by defining an action strategy. Various alternatives can be adopted, such as removing the partial information, replacing the missing parts with non-indicative values, inserting an estimation based on the existing data, or logical reasoning.

Another quality to check is data consistency. For example, ensure that the values all have the same units of measurement. Similarly, we must remove outliers, duplicate values, and measurement errors since they only generate noise and are not valid for statistical purposes.

Finally, when large amounts of data are available, it is convenient to apply filters to eliminate redundant information that could slow down operations and waste data storage space.

The last phase of the data supply chain is the *Loading* one, actually when loading data in a *data warehouse*. For large quantities of data, the loading operation can consume time, even if not complicated. The data warehouse could have different Database Management Systems (DBMS), distributed and redundant for security and efficiency reasons. Although most popular DBMS can be adapted, it is usually more efficient to store data in appropriate databases, according to their nature. For example, the Relational

SQL Database better handles data with specific relations. Data with a document nature is better managed in NoSQL databases, offering a high level of flexibility. Semantic data, instead, is better governed by Graph databases.

## 5.5 LIVING LABS AND OPEN INNOVATION: A PROTOTYPE

Aggregating data that flow across individuals and the surrounding territory produces knowledge to support the decision and ultimately deliver it back to individuals as services. Such a conceptualization also fits Flichy's model of contemporary society named "connected individualism", which emphasizes the role of communities and networks for individual growth and empowerment [58]. Thus, developing a combination of territorial knowledge with personal data and events, that are organized through a software infrastructure conceived for special-purpose applications, meant to improve users' experience. In addition, creating public value for services can achieve the goal of information sharing and cooperative exploitation.

To design and develop such a platform taking into account the need for a shared communication protocol among all the involved entities, extensibility (i.e., the ability to add new features without affecting the existing components), and the opportunity to hide the format differences of data coming from heterogeneous data sources is of primary importance. In particular, for this aim, it is necessary to identify people who produce information, to aggregate/disaggregate it, and to include slices of them as part of the system, thus satisfying the need to access and provide both prior knowledge (static) and information generated in real-time (dynamic).

The Web platform described in this section supports citizens' activities and allows service providers to create, monitor, and share different services. It represents proof of the expressed concepts. The whole community may leverage such services and the derived (possibly aggregated) information, with potential growth in territorial knowledge.

The proposed application creates a single platform to apply most of the phases previously described, starting from the data extraction to the visualization of the analysis results. It is possible to generate maps of the local territory and identify trends and phenomena occurring in one or more municipalities. It is also possible to quickly and intuitively group geographically distant municipalities sharing the same characteristics.

The goal of the scenario used to run the prototype was to develop and implement policies to promote the economic regeneration of local areas. In particular, the knowledge derived from the platform and visualized through the maps has become fundamental support for expert users, such as urban planners and engineers, to understand which Smart Specialization Strategy is best suitable to stimulate local economic development.

In terms of data, the first source in terms of municipalities, census sections, provinces, and regions is the ISTAT. However, despite providing certified, verified, and reliable data, the ISTAT portal is quite complex to build the desired queries, even for experienced users. Out of this concern came the development of a simplified application. It allowed direct requests to be made through a faster and more natural interface, thanks to a technologically advanced system that managed the ISTAT data warehouse in machine-to-machine mode via web service. This service, free of charge and freely available, allows bodies and organizations to formulate specific requests, download data of interest, and quickly receive them in their databases, web portals, and applications [64]. The queries can be sent through programming languages, like PHP, C#, JAVA, and VB.NET. In contrast, the query formulation follows the Statistical Data and Metadata eXchange (SDMX), adopted to exchange statistical data by the most important international organizations.

It is possible to query web services by selecting and inserting information, such as the category and the territory of interest at the regional, provincial or municipal level. Then, the expected results are displayed on the map and in tabular form. To apply the georeferencing algorithm, data transformation requires a supervised action where the user manually indicates which fields contain the municipality or census section code to which data refers.

In addition to data loading and map visualization, the proposed application performs statistical functions, data analysis functions, interactive diagrams, and composite indexes to identify specific phenomena or deficiencies in a given territory.

An example of a composite index building is the application of the Mazziotta-Pareto Index (MPI) [86]. It summarizes a set of individual indicators assumed to be non-substitutable, i.e., all components must be balanced [33]. MPI + is usually chosen for "positive" multidimensional phenomena (the higher the value, the better the performance), such as well-being, quality of life, development, and infrastructure provision. In contrast, MPI - is chosen for "negative" multidimensional phenomena (the higher the value, the worse the performance), such as poverty, mortality, and pollution. Figure 5.4 displays the trend of the composite index representing the combination of socio-economic variables on a municipal basis.

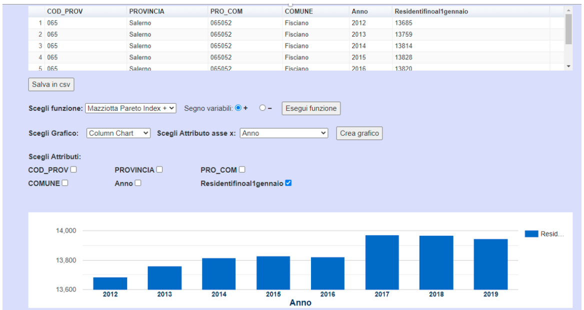



Figure 5.4: Mazziotta-Pareto Index application (Source: [7])

It is also possible to execute the Principal Component Analysis, whose purpose is to reduce the greater or lesser number of variables describing a data set to a smaller number of latent variables, limiting the loss of information as much as possible [133]. When combined with the K-means algorithm, this technique allows for building clusters, which minimize the total intra-group variance [82]. Figure 5.5 shows an application of these two combined analyses, where the resulting clusters represent homogeneous groups of municipalities for the given socio-economic variables.

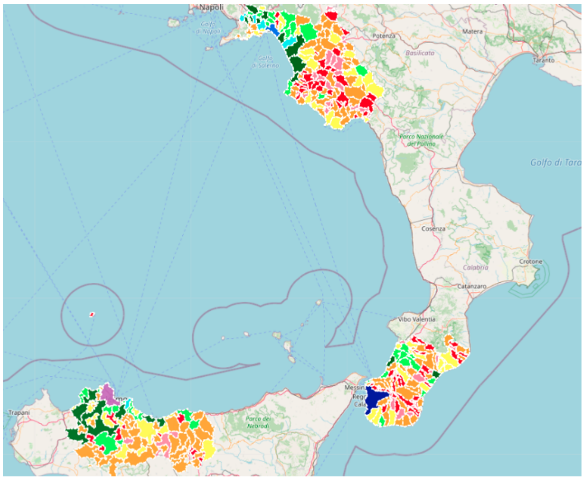



Figure 5.5: Clusters of municipalities for socio-economic parameters (Source: [7])

## 5.6 REMARKS

This chapter proposed an architecture and a software application to re-think LL activities to improve the citizens' engagement and produce knowledge dynamics as is envisaged by the Strategies for Smart Specialisation (S3). The work investigated the technologies to extract, transform, and load Open Data. The goal was to facilitate contextualized open innovation processes involving the citizens and other territorial stakeholders. The tested platform helped to understand the operational and technical difficulties, related to the integration of government Open Data and the analysis tools made public, leading to the following two main lessons:

- Open Data collection by the government agency is not as easy as expected.
- A rich platform, although appreciated by experts in the domain, can be difficult for non-expert citizens to use.

# 6 INTERACTIVE MAPS

## 6.1 SUMMARY

Interactive maps represent an easy and effective way to visualize geographic data, and this chapter enhances them by experimenting with new visual means. This chapter describes the Chorems as a visual synthesis of dynamic phenomena by introducing a use case. Furthermore, it is described how the use of Chorems could simplify the analysis of a phenomenon as Urban Heat Island (UHI), and new visual representations are introduced to represent the components affecting the outline of UHI.

## 6.2 INTRODUCTION

Nowadays, the topic of smart cities is increasingly central in the dynamics of management, transformation, and redesign of modern cities, and is strongly related to the concept of smart communities. In [109], the authors synthesize a smart city as *"a city exploiting IT services to connect people to each other, to city services, infrastructures and organizations with the goal to create a common conscience or knowledge that can improve the community life of the same city"*. Speaking of smart community, instead, it is defined as a *"community in which government, business, and residents understand the potential of information technology, and make a conscious decision to use that technology to transform life and work in their region in significant and positive ways"* [131]. Although in both those contexts, technology is imagined as the main objective [128], many researchers and local administrators prefer to put the citizen at the center and consider technology as mere support for the community to share common visions and reach common goals. Such an approach is in line with the actualization of the OG's three pillars, namely transparency, participation, and collaboration [93]. They motivate the importance for a smart city administration to share decisions with citizens to receive real support and legitimize their decisions.

As explained in [89, 128], a way to engage citizens is to make available government data as open. Open data allows the administration to achieve higher transparency in decision-making and encourages the citizens to support the strategic objectives building a network that improves the individual and collective quality of life. However, data and information used to make administrative decisions and involve citizens can sometimes represent complex territorial phenomena and processes, requiring specific knowledge to interpret. Problems may arise, in particular, when data and information are related to phenomena that are implicitly complex by nature. In these cases, several and different parameters concur, which can also have various weights depending on the evaluation scale.

This chapter aims to show the effectiveness of *Chorems* as a communication and support tool for understanding the dynamic realities of a territory. When maps of *Chorems* visually synthesize the urban contexts, they can produce a territorial schematic representation, which eliminates details not functional to the map comprehension.

In particular, to show the efficacy of *Chorems*, this chapter proposes them to represent the complex phenomenon of UHI.

Below is illustrated the work done in collaboration with researchers of the Civil Engineering Department and the Geographic Information Systems Laboratory of Computer Science Department of the University of Salerno, published in [14].

The phenomenon of urban heat islands is invisible but typically present in most modern cities. It is composed of different elements present in the urban territory, and its understanding implies technical and scientific knowledge not common among citizens and administrators. However, it is a phenomenon affecting all of us daily. Therefore, to explain how to manage the determinants of the phenomenon and reduce its impact, a visual paradigm must be identified. It must be easily understood and usable by decision makers who need to interface with technical personnel and easily usable by citizens who want to understand the reasons for the actions taken.

Maps of *Chorems* can represent a solution. They offer a synthetic global view that is suitable to guarantee such requirements.

## 6.3 CHOREMS AS VISUAL SYNTHESIS OF SPATIO-TEMPORAL PHENOMENA

The concept of *Chorem* was first introduced in 1986 by the French geographer Brunet as a schematic territory representation, which eliminates details not functional to the map comprehension [23]. Since then, the concept has evolved over the years, but *ad hoc* solutions fail to convey the information as intended, because of the lack of a rigorous approach to the creation and composition of *Chorem* [4, 6, 100].

Figure 6.1 shows a set of *Chorems*, including industrial zones and big poles, which make up a thematic map referring to Spain.

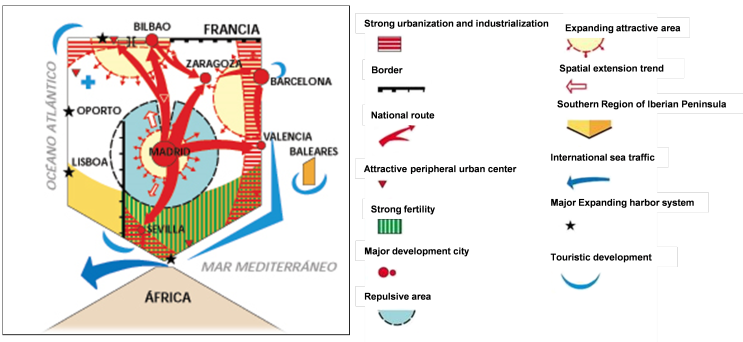


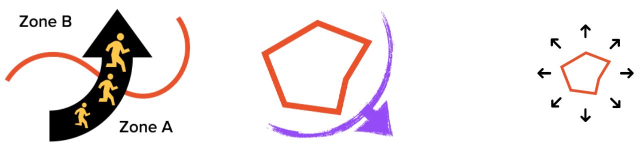

Figure 6.1: A Map containing *Chorems* about Spain and its legend. (Source:[36])

A definition and classification of *Chorems* as visual syntheses of geographic database contents was first provided in [36], to homogenize *Chorems* construction and usage and to provide a usable framework for computer systems.

According to it, *Chorems* fell into three main categories, namely:

- Geographic *Chorems*,
- Phenomenal *Chorems* and
- Annotation *Chorems*.

Geographic *Chorems* represent geographic data with associated simple geometries, such as points, lines, polygons, and objects made up of their combinations, such as networks. Phenomenal *Chorems* describe spatio-temporal phenomena involving one or more geographic *Chorems*, and, when useful, they can be further classified in Flow, Tropism, and Spatial Diffusion. A Flow *Chorem* represents objects movement between geographic *Chorems* (Figure 6.2a). A Tropism *Chorem* represents a homogeneous attractive or repulsive space, around a geographic *Chorem* (Figure 6.2b). It can represent phenomena that tend to be repelled or attracted from/to a specific area. A Spatial Diffusion *Chorem* represents a spatial progression or regression, from a geographic *Chorem* along a given direction (Figure 6.2c). It can represent a phenomenon that varies its area of influence. Finally, an Annotation Chorem represents map labels or remarks, useful to provide users with additional information about the map.




(a) Flow (b) Tropism (c) Spatial Diffusion

Figure 6.2: Phenomenal Chorem classes examples

Figure 6.3 depicts the underlying structure of a *Chorem*, which was first described in [36]. It considers the complex nature of geographic data and phenomena by visually integrating the iconic and the property components. As for the former, the iconic representation assemblies a graphical component, corresponding to the visual representation, and a meaning, referring to the semantic component. In such a way, users can quickly perceive the meaning associated with data and use it properly. Two parts compose the property component of a *Chorems*, a type attribute specifying the category the *Chorems* belongs to, namely geographic, phenomenal, or annotation, and a source indicating where data could be retrieved (such as a table or a view name, a SQL query or a function). It is worth noting that, in the case of a phenomenal *Chorems*, the type attribute also contains information about the geographic *Chorems* to which it is related.

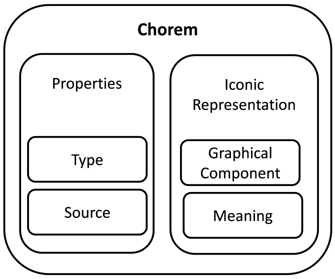



Figure 6.3: The structure of *Chorems* (Source:[36])

The concepts of *Chorem* element, *Chorem* and *Chorematic* Map were introduced as follows:

- a *Chorem* element is a basic element representing either a single geographic object, such as a city and a road, or a single phenomenon, such as people migration and car mobility;

- a *Chorem* is a set of homogeneous *Chorems* elements. For instance, the set of the most important Italian cities or the set of the main flows between such cities.

- a *Chorematic Map* is a set of *Chorems*, that schematize data of interest related to a specific place or region. A legend can be associated with a *Chorematic map*, which explains the meaning of each *Chorem*.

This classification emphasizes the relationships existing among a single instance of a *Chorem*, the group to which it belongs, and the whole scenario to arrange spatially different *Chorems*.

Laurini et al. in [35] discussed the potentiality of *Chorems*, giving a list of the different roles that *Chorems* may play. The authors state that *Chorems* can be used to represent geographic knowledge, to visually summarize database contents, and finally to underlie the creation of a novel entry system for geographic databases.

Starting from these definitions, in [32, 36], the authors demonstrated how *Chorems* could represent the starting point for further processing tasks aimed to derive spatial analysis data and to support expert users in decision making. They introduced a set of geographic and semantic operators to derive spatial knowledge helping domain experts to face and get rapid and exhaustive responses in critical situations.

Based on Ben Shneiderman's mantra and Keim's adaptation to the Visual Analytics domain [73, 115], applying these operators allows users to navigate a chorematic map from an initial overview to detail, as follows.

*Geographic Zoom*

This operation corresponds to the traditional map zooming (in/out) operator. When applied to a *Chorematic* map, it acts exclusively on the visual aspect of the *Chorem* by changing the size of the visible details of the involved *Chorem* elements, leaving them unchanged, also in terms of structure.

*Semantic Zoom*

Generally speaking, a semantic zoom (in/out) changes the type and meaning of information associated with *Chorems*. It allows access to a different level of information. It analyzes the *Chorem* and its elements by (dis)aggregating and visualizing them in detail. In particular, when applied to geographical *Chorems*, they just split up (resp., aggregated) by showing a new level of geographic data abstraction. The *Chorem* structure changes accordingly, substituting the corresponding iconic representation component in terms of graphical component and meaning.

The application of a semantic zoom to a Phenomenal *Chorem* decomposes (resp., aggregates) along with the Geographic Chorems related to it, the Chorem itself. In this case, the *Chorem* structure changes differently. Even if the output *Chorem* still corresponds to the initial phenomenal *Chorem*, its meaning changes, referring to a different abstraction of the territory.

*Geographic Filter*

This operation allows the user to select Phenomenal Chorems elements by using the graphical component of one or more Geographic Chorems elements as a spatial filter. In this case, the condition corresponds to the territory of interest where phenomenal *Chorems* elements have to be analyzed. The output of the operation consists of one or more phenomenal *Chorems* that satisfy the condition. The resulting *Chorem* structure does not change in terms of properties and iconic representation, whereas the number of phenomenal *Chorem* elements may vary to satisfy the condition.

*Semantic filter*

This operation allows users to filter *Chorem* elements that satisfy a particular condition, by directly operating on their semantics.

In order to show how the visual representation based on *Chorems* can be useful to handle the information associated with the UHIs, the following Section briefly describes this environmental phenomenon and presents a model to process a set of parameters associated with it.

## 6.4 THE URBAN HEAT ISLANDS PHENOMENON

The increase in mean temperature between public urban areas and surrounding rural areas determines a thermal anomaly known as UHI [106]. This phenomenon can have significant differences between cities with a comparable population and varies considerably according to weather conditions, season, and time of day. Its most relevant effects are:

- an increase in energy consumption,
- high emissions of pollutants and so-called greenhouse gases,
- a worsening of the life and thermal comfort of the population, and
- an increase in storm phenomena in the urban area.

These effects significantly affect outdoor air quality and make the UHI phenomenon relevant to improving urban life quality. So to achieve this improvement, it is necessary to consider that urban areas are more energy-rich than rural areas, and this imbalance is even more significant as the number of heat sources increases. The factors that influence a UHI fall into three groups:

- factors related to the environmental context;
- factors related to the human component;

- factors related to the urban structure.

The first two components are defined in [106] as uncontrollable. In contrast, the causes linked to the anthropic modification of the territory, defined as controllable variables, are urban geometry, properties of urban materials, and urban green.

The urban geometry contributes to the UHI phenomenon through the so-called urban canyons, consisting of the high building walls that face each other along the streets. Here, the solar radiation is reflected several times from the road surface and the walls of the buildings, remaining trapped inside the canyon and causing a rise in temperatures. The materials used in urban areas modify the energy balance mainly through the albedo, which indicates the fraction of incident radiation reflected off a surface and varies considerably concerning the color and type of surface. Indeed, the materials present in cities have a lower albedo than those in the countryside. The presence of green areas, on the other hand, offers a mitigating contribution to the phenomenon. The crowns of the trees can shield the solar radiation and use a part of this for photosynthesis.

Based on the analysis of these factors, it is possible to identify specific mitigation actions, i.e., interventions that should:

- modify the morphology and adopt materials with lower thermal admittance, thus reducing the flow of heat stored in the urban structure;
- increase the albedo of the surfaces and modify the geometry of the buildings, thus reducing the net radiation;
- increase the permeability of surfaces and the presence of vegetation;
- reduce vehicular traffic and improve energy efficiency, thus reducing the flow of anthropogenic heat.

The inner complexity of the UHI phenomenon implies a selection of visually analyzed components related to the investigation scale of the phenomenon. In [61], the authors studied more than 250 scientific articles on urban heat mapping indicators and mapping techniques. They found that one hundred thirty-three articles did use a cartographic map to indicate UHIs. However, although it is popular visually analyze the UHI phenomenon with mapping, it is still challenging to represent a single complex indicator derived from multi-variate data. The Chorems paradigm, with its geographic and semantic operators, appears to be a valuable mechanism for a more immediate understanding of the causes of such phenomenon's complexity at any scale.

In the following Section, the controllable variables are modeled in terms of *Chorems*. They are used to visually express the basic features of the critical zones identified by a model underlying the structure of the UHI *Chorems*.

## 6.5 CHOREMATIC MAPS TO VISUALLY SYNTHESIZE AND ANALYZE URBAN CONTEXTS

By representing the previously mentioned elements through proper *Chorems*, it is possible to design and build *Chorematic* maps to support decision-makers in monitoring activities to improve citizens' lives.

The goal of this Section is twofold. First, describe a set of *Chorems* specifically conceived to visually synthesize an urban context and some relevant phenomena and elements related to it. Then shows how geographic and semantic operators can build *Chorematic* maps to investigate the UHI phenomenon.

### *Chorems for the urban context*

Figure 6.4 shows the list of *Chorems* associated with the UHI phenomenon and each factor contributing to its development, namely *Urban Area*, *Urban Greenery*, *Urban Geometry*, and *Albedo*. They are structured as described in Figure 6.3 and, when applicable, a classification of their possible ranges is given. The geographical context *Urban Area* represents the visual synthesis of an urbanized area to which different phenomena can be associated. As an example, will be considered the instances of districts and quarters.

The *Urban Greenery* geographic *Chorem* represents the visual synthesis of public landscaping and urban forestry created to let citizens benefit from environmental green spaces. Arbore-

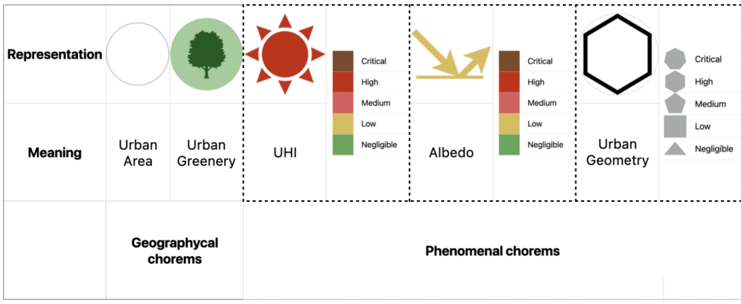



Figure 6.4: Chorems for the UHI phenomenon (Source: [14])

tums, parks, and urban vegetable gardens are instances of *Urban Greenery*.

Associated with the *Urban Area Chorem*, *Urban Geometry*, and *Albedo* are Spatial Diffusion *Chorems*. The *Urban Geometry* source is related to the average complexity of the urban area, which considers some spatial indices, such as size, shape, skyline, and openness of the area of interest. A polygon represents it, whose number of sides ranges from 3 (negligible geometry) to 7 (critical urban geometry), thus increasing with the criticality performed.

As for the *Albedo Chorem*, its iconic representation is similar to a *tick* made up of two arrows, whose color shows the capability of the associated area to absorb/reflect the solar light.

Finally, UHI is a Spatial Diffusion *Chorem* associated with the *Urban Area Chorems*. The UHI source corresponds to the calculation performed on the factors as defined in Section 6.4. Its iconic representation is a *sun* whose color intensity indicates the level of criticality.

In addition to the above *Chorems*, a set of phenomena, such as the area's average temperature, pollution, wind, traffic, and the quality of the urban greenery whose presence and intensity can affect one or more factors underlying the UHI phenomenon, characterizes the urban context. Then, to derive a complete scenario of the urban context, it is necessary to associate a *Chorems* representation with each of them. This approach allows decision-makers to obtain a global overview of an area of interest and analyze every single object and phenomenon featuring it.

Figure 6.5 collects *Chorems* modelled to this aim, namely *Pollution*, *Winds*, *Traffic*, *Temperature*, and *Urban Greenery Quality*. They are Spatial Diffusion *Chorems* associated with the *Urban Area Chorems*. In particular, *Pollution*, *Winds*, and *Traffic* represent elements that can affect the temperature, and their sources are the sensors located in the area of interest, along with sensors for temperature. A numeric annotation enriches the iconic representation of temperature that shows an arrow directed clockwise (resp., counterclockwise) when the average temperature measured in the area is higher (resp., lower) than the city temperature average.

Finally, the *Urban Greenery Quality* indicates the quality of the vegetation in the depicted area, which can depend on the seasons and the care of the plants. The high quality of the urban greenery alleviates the impact of the heat island. Its source is a database frequently updated with a quality index ranging from 1 to 10 and estimated with data from satellite images or field inspections. Two concentric circles containing a tree are its iconic representation. The outer circle represents the area under examination. The inner one is the extension of green related to the region under assessment, while its color intensity is directly proportional to the green quality index.

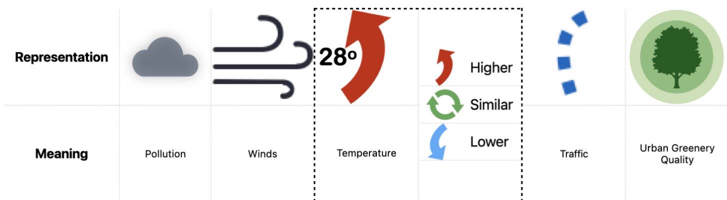



Figure 6.5: Chorems affecting the UHI phenomenon (Source: [14])

*Analysing the UHI phenomenon*

A chorematic map can emphasize essential aspects of a territory, which are valuable for a better comprehension of the visualized phenomena. Moreover, it offers a synthetic global overview, further investigatable, to analyze specific spatiotemporal information built upon large sets of source data, otherwise difficult to manage. In particular, by zooming and filtering tasks in terms of geographic and semantic operators, users may gradually reduce the search space and select a subset of data compliant with

Shneiderman's mantra and Keim's interpretations of the visual information-seeking paradigm. Then, users can select and query a *Chorems* element to obtain descriptive information related to it.

Figure 6.6 (on the left) shows a chorematic map containing seven UHI *Chorems*, different in intensity, calculated in seven different zones of the city.


Figure 6.6: A geographic zoom-in (Source: [14])

Chorems visually summarize the outcome influenced by the components in that area of the city that contribute to the UHI. A geographic zoom-in derives from the scenario depicted on the right.

The application of a semantic zoom-in to the red UHI *Chorems*, disaggregate it according to the parameters described in Section 6.3, while the given scale still holds, as shown in Figure 6.7.

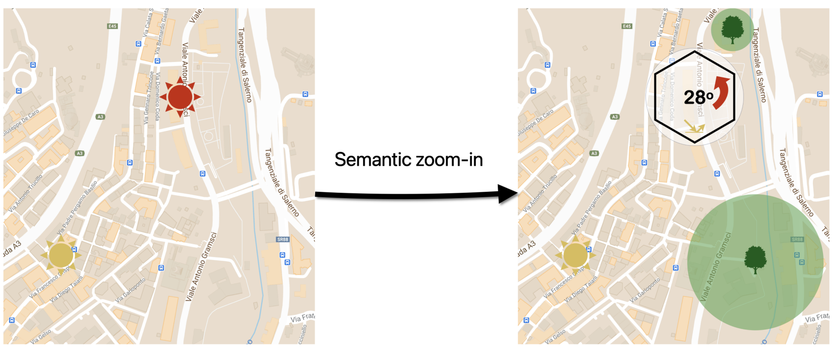



Figure 6.7: A semantic zoom-in (Source: [14])

It shows the involved *Chorems*, namely *Urban Area*, *Albedo*, *Urban Geometry*, and *Urban Greenery*. The area represents the extension of the UHI phenomenon. The color of the *Albedo Chorems* indicates the medium level of the albedo in that area. The grade of average urban geometry is high, being represented by a hexagon. The presence of urban greenery is indicated by two *Chorems*, differently sized to express their different impact on the heat island. Finally, the red arrow indicates the ratio of the temperature reported in the area with the temperature average of the city, along with the numeric temperature average.

Figure 6.8 shows a semantic filter applied to derive UHIs whose criticality is higher than "Low".

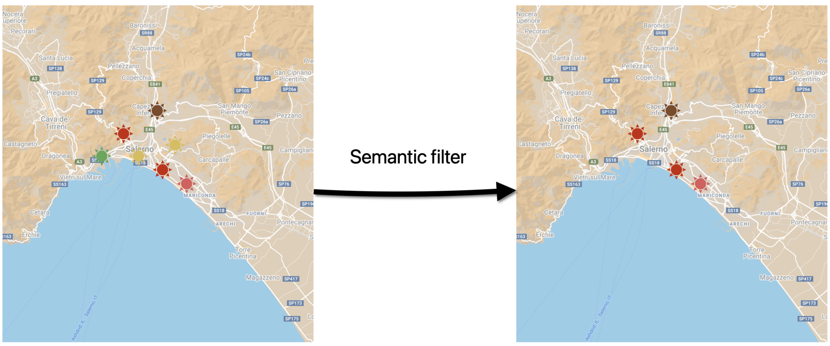



Figure 6.8: A semantic filter (Source: [14])

Figure 6.9 displays a *Chorematic* map where instances of each type of *Chorems* previously introduced occur for the UHI *Chorems* under investigation.

Here, proper sensors distributed on the territory acquire data about lines of traffic, air pollution, temperature, the presence of winds, and the quality of the urban greenery.

Finally, Figure 6.10 shows a *Chorematic* map resulting from a semantic zoom-in applied to two UHIs.

This map allows users to investigate the phenomenon occurring in two different areas by comparing the corresponding factors in the form of *Chorems*.

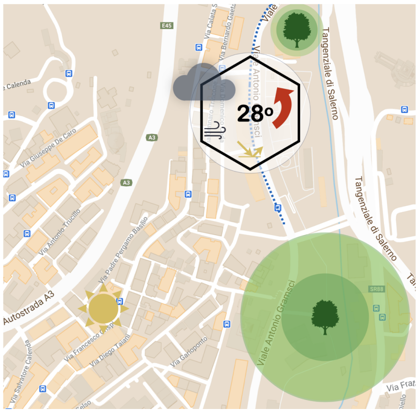



Figure 6.9: Chorems in a urban context (Source: [14])

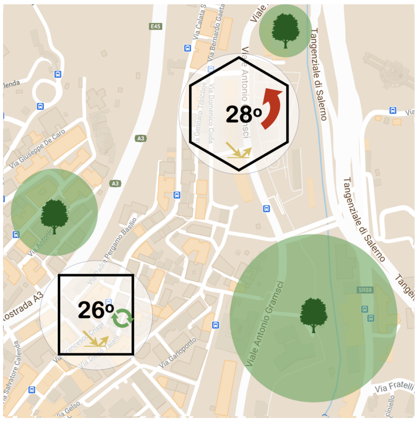



Figure 6.10: Comparing UHIs (Source: [14])

## 6.6 A CHOREMS-BASED APPROACH TO GET CITIZENS INVOLVED IN DECISION MAKING

This Section describes how decision-makers and citizens can take advantage of chorematic maps in investigating urban contexts and understanding environmental phenomena impacting them. It describes two scenarios in which designing plans and communicating them to citizens through a simple visual language based on the expressive power of *Chorems* can support decision-makers.

*Identify a portion of territory on which to rethink the urban greenery*

Today, political decision-makers design and plan the installation and renovation of urban green parks and tree-lined avenues, often considering them as street furniture or leisure areas instead of strategic urban elements. Various criteria come into play in these design activities, such as the cost of work, the possible complications for mobility, and the possibility of recovering degraded spaces. Moreover, other criteria, such as the urban heat islands, even equally relevant for their impact, are often not sufficiently considered. The availability of simple methods and tools to study complex phenomena could mitigate this discrepancy and contribute to understanding the factors influencing them.

A *Chorems*-based approach can represent a solution to bridge the gap. Decision-makers, who aim to explore city areas, can adopt *Chorematic maps* and analyze specific contexts to identify complex phenomena, such as UHIs in the urban area, without any effort. Their color represents the criticism of each UHI, and decision-makers can apply a semantic filter to identify those areas whose criticality overcomes a given threshold. Then, for the reorganization of the urban greenery, the decision-makers can consider those areas afflicted by the phenomenon more carefully. Moreover, a semantic zoom-in on a selected UHI allows decision-makers to know the extension of the critical area and the presence of the greenery nearby, whose level of quality can be observed, in turn. Hence, the adopted *Chorematic map* simplifies and highlights the comprehension of the UHI phenomenon and facilitates its use as an additional criterion to identify the portion of territory where intervening.

Finally, adopting *Chorematic maps* as a communication tool can be valuable for sharing the planned urban works, particularly the urban green redesign plans, with citizens. A map can support citizens' understanding of the project by showing them how the proposed interventions are reasonable to address the phenomenon of UHI. Thus improving the welfare of the entire community and avoiding general regret due to the lack of understanding of why planned interventions affect one area of the city rather than another.

*Designing targeted interventions for the prevention of heatwaves to protect the health of the elderly and children*

During the summer, the elderly and children can be severely prone to heatwaves, which can cause serious health risks. The municipal administration arranges for the deployment of ambulances and mobile medical points in various areas to quickly bring help. They can also set up points for the administration of water bottles and arrange sun covers in the avenues. Finally, they may want to communicate to the population to stay at home and avoid some particularly affected areas. Making such a plan involves a great waste of time and resources.

Based on *Chorematic maps*, the city administration can easily identify the areas of the city most prone to the UHI phenomenon by selecting the most serious ones through a semantic filter. Moreover, areas at risk with sensitive sites can be detected by applying geographic and semantic zooms, such as schools, nursing homes, parks, and squares. This approach allows the municipal administration to allocate its resources in the best way.

As for citizens, in order to let them know the resources distribution, it is possible to set up public totems in strategic areas, such as bus and subway stops, where running a *Chorematic map* shows the main UHIs and the corresponding temperature increment. This action can help citizens understand the gravity of the phenomenon and plan their routes and activities for the rest of the day, avoiding the riskiest areas.

## 6.7 REMARKS

By performing a map overview and querying tasks is possible to obtain a global view of a scenario featuring a domain of interest. Since using visual and interactive methods have the risk of getting lost inside the original data collection when dealing with a large amount of them, it is reasonable to apply first some analysis and then provide an overview of the resulting relevant contents and details on demand. In this way *Chorematic maps* for urban contexts act. They visually summarize relevant objects and phenomena of an area of interest and then provide users with valuable information to better understand what characterizes a scenario.

This chapter adopted the *Chorems* to describe the UHI phenomenon and the factors determining it. The model used to process those factors and calculate the extension of the critical zone is a "work in progress" from a physics point of view. Nevertheless, possible improvement can be reconsidered in terms of *Chorems* thanks to the level of abstraction of their structure, which expresses each component in a way abstract enough to be independent of specific implementations. This feature also paves the way for the use of *Chorems* to build "what if" scenarios where the interaction addresses the simulation of the presence/absence of phenomena and the modification of parameters to evaluate possible outcomes.

# 7 ACTIONABLE CHOREMS

## 7.1 SUMMARY

In this chapter are described the new concepts that this thesis wants to introduce. The information discovery and analysis it is enabled and facilitate through the creation of new kind of *Chorems*. The traditional *Chorems* has been transformed into a new tool that, when used actively and dynamically, allows for new analyses of the data shown and the discovery of previously undisclosed information.

## 7.2 INTRODUCTION

Chapter 5 investigated the use of LL as a platform to worthily share local knowledge. It suggested tools to acquire and analyze territorial data in ways that effortlessly help decision-makers and policymakers. Although LLs aim to involve administrators, experts of the domain, and citizens, the presented tools could still be challenging to use by a non-expert citizen.

In Chapter 6 the *Chorem* and *Chorem maps* were introduced to offer more straightforward tools for analyzing data related to geo-referenced phenomena, even if complex and dynamic. Although literature widely proposes Chorems to visually present results of spatial data analysis, [32, 36, 57, 76] even about evolutionary phenomena, since the Chorems were born for cartography on paper, no works leverage all the current technologies and methods of Human-Computer Interaction (HCI).

The present Chapter 7 describes the research aiming to leverage HCI to create interactive, intuitive, and easy-to-use visual analytics tools for Big Data for Sustainable Development.

Proposing an evolution of *Chorems* on computer displays, they are transformed from passive visual analytical tools into an actionable tool. In this way, users and applications can perform actions on *Chorems* themselves, achieving new features.

## 7.3 CHOREMATIC DISPLAYS

Although *Chorematic maps* can synthesize information without a geographic map, placing *Chorems* on it yields more knowledge derived from the additional features of the place that a map visually represents. For example, a *Chorematic* map of UHI phenomena in various cities, can be fully represented with *Chorems* only, but adding a geographic map reporting the elevation data of places could bring out a new relation between the phenomenon and the place where it is measured, not previously detected and depicted.

As seen in Section 6.3, the relationships existing among a single instance of a *Chorem*, the group to which it belongs, and the whole scenario to spatially arrange different *Chorems*, are defined and fixed at the creation.

The users can apply some well-defined operations on the map, namely *Geographic Zoom*, *Semantic Zoom*, and *Filtering*. These actions will produce a new map with the same information at a different scale or filtered. It is worth noting that these actions are applied on a *Chorematic map*, not on *Chorems* or their components. Indeed, the *Chorem*, as defined, shows a synthesis of information obtained by queries conceived by map creators and does not permit the discovery of new information. Thus, the users can only disclose predefined information.

A step forward is proposed in the following, where the *Chorems* can be used as a visual tool to perform user-defined analysis. The users can explore and discover even information not initially conceived by the map creator.

To achieve this aim, the *Chorem* moves from an almost static visualization tool to an actionable tool with a dynamic iconic representation, i.e., an object on which it is possible to take some actions to obtain new analysis.

Since this paradigm shift can be conceived only on a computer display and not on a printed map, the *Chorematic Map* is redefined as a *Chorematic Display*. The term *Chorematic Display* would emphasize the centrality of HCI.

*Moving from representation to exploration*

Figure 7.1 shows the *Urban Area Chorem* and the *Urban Greenery Chorem* described in Chapter 6.

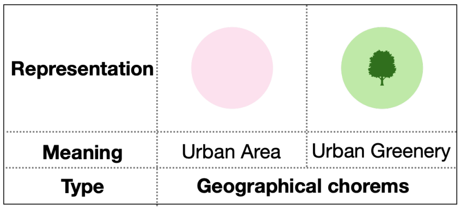
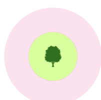



Figure 7.1: Area and Greenery Chorems described in Chapter 6

By combining the two Chorems, it is possible to visually represent the greenery in a specific geographic area. Figure 7.2 illustrates how the combined *Chorem* presents the quality, the quantity, and the kind of green in an urban area. The central icon represents the type of green, whether forest, gardens, mixed, or other. The green circle diameter of *Urban Greenery* shows the rate between the geographical area considered and the percentage of greenery covering it. The intensity of green color in the circle of the *Urban Greenery* is proportional to the quality of the green, such as the Normalized Difference Vegetation Index (NDVI) [116], assessed over the extent of the *Urban Area*.



(a) 50% coverage of low green quality. *Forest* type

(b) 90% coverage of medium green quality. *Garden* type

(c) 70% coverage of high green quality. *Mixed* type

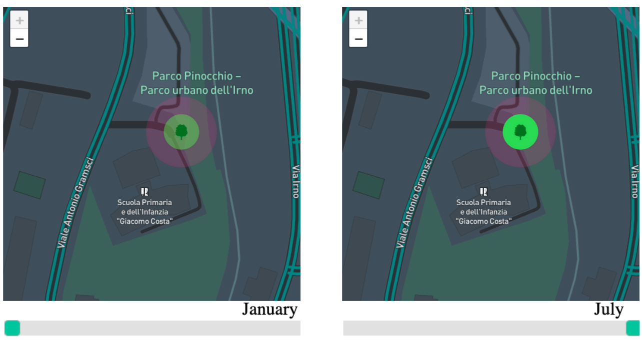
Figure 7.2: Urban Greenery Chorem on a Urban Area Chorem

Figure 7.3 shows an example of the previous *Chorem* on a geographic map.


Figure 7.3: Chorem in a urban context

As it turns out, just a few visual symbols can summarize much information.

Since *Chorems* are on a computer display, they can dynamically change their visual representation. For example, showing the map during the winter and the summer months will fetch different information, and the *Urban Greenery Chorem* will slightly change its representation. Figure 7.4 shows an example where the color intensity of greenery will be greener in summer than in winter because the NDVI index varies according to the season.




(a) Winter Time (b) Summer time

Figure 7.4: Chorem in a urban context

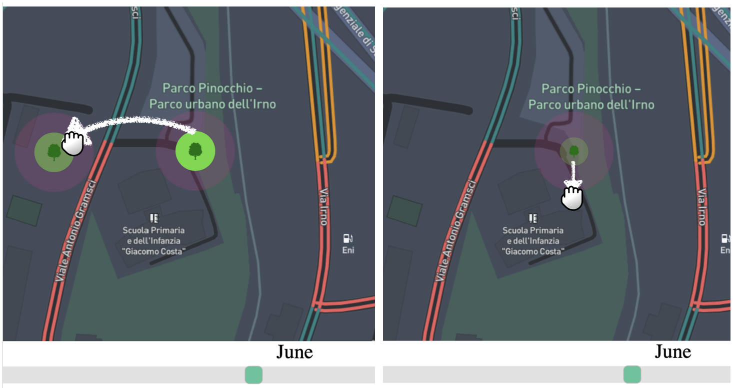
The above example shows one of the possibilities that *Chorem* offers when on a computer display. It allows exploring the information temporarily.

By animating the map with a sequence of snapshots taken at different times, it is even possible to visualize the trend of urban greenery.

Figure 7.5 shows two other examples of how to spatially explore the information by *Chorem*.

Figure 7.5a shows the *Chorem* dragged from one location to another and how its representation changes due to the location change. Moving the *Chorem* to another extent modifies the information belonging to it and consequently its representation.

Figure 7.5b shows how enlarging the extent of the *Urban Area* affects the *Urban Greenery* representation since a more extensive area on which to evaluate the greenery will produce different information.




(a) Dragging Chorem (b) Stretching Chorem

Figure 7.5: Chorem transformation by user

The activities presented above are of two kinds, actions that affect the data and actions involving the representation. Moving the slide bar on the map modifies the data query, and consequently, the *Chorem* representation changes. Enlarging the circle of *Urban Area Chorem* affects a part of the Chorem representation, generating a new query on the different geographic extent. It modifies the result set and then the related *Chorem* representation (*Urban Greenery Chorem*).

Since the *Meaning* component of the *Chorem* binds the information with the representation, changing one of them induces a change in the other, keeping the semantic meaning consistent.

Conceptually, the *Chorem* can receive external stimuli inducing reactions. The reaction always produces a new data set, and the *Chorem* must visually represent the new information according to the semantic meaning.

The reaction can have two directions:

- a modification in the graphical representation of the *Chorem* determining a new data set;
- a change in the data set determining a replacement of graphic representation.

The above examples show how the user can act on the GUI and explore the information spatially and temporarily. Although HCI produces most of the events for the *Chorem*, they must not be limited to it. Indeed, the *Chorems* can also catch external events, as shown in the example below.

Figure 7.6 shows the *Pollution Chorem* introduced in the previous Chapter 6. The *Pollution Chorem*, in this example, is the *Phenomenon Chorem* of pollution having the European Air Quality Index (EAQI)[53] calculated on data produced by a set of stations.

| Representation |  |  |
|---|---|---|
| Meaning | *Urban Area* | *Pollution* |
| Type | Geograpohical chorem | Phenomenon chorem |

Figure 7.6: Pollution Chorem

Combining the two *Chorems* of Figure 7.6, it is possible to visualize information about the pollution of a geographic area defined by the boundaries of the *Urban Area Chorem* instance (Figure 7.7).

Since the EAQI unites information of five different air pollutants to show the current state of air quality, the index depends on a set of sensors installed on a group of stations. The stations transmit data with different duty cycles, raising the issues already discussed in the first part of this thesis. In addition, they may

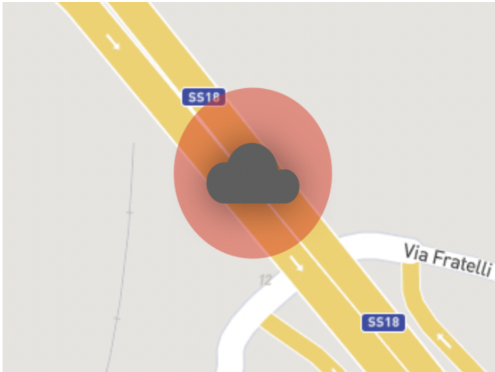



Figure 7.7: Urban area and Pollution Chorems.

be subject to unexpected malfunctions, scheduled maintenance and calibration, or any other situation that makes their data unavailable for some period, causing a less responsible index.

Having information about the condition of devices helps to estimate their liability. To do so, sensor stations or those who manage them could broadcast messages to all affected data consumers, alerting them about the station status.

The situation described above is an example of external events that could activate changes in the graphic representation of the *Chorem*.

For example, if an external application broadcasts an alert message, all data consumers will receive it. From *Chorem*'s point of view, the reception of an alert message is an event to which a reaction follows. The reaction consists of a change in representation and a new data set that excludes unreliable data from the analysis. Thus, users receiving correct information are immediately alerted of its incompleteness. The following Figure 7.8 is an example of such a *Chorem* representation on a map.

Since the EAQI has some color to express the level of the air quality [54], the color of the *Pollution Chorem* will be compliant with it. A Legend will appear whenever the mouse cursor is over the *Chorem*. The alert icon on the cloud symbol notifies the user about the incompleteness of the data.


Figure 7.8: Example of Pollution Phenomemon with an alert messages

*Chorematic Display concept*

It follows an extension of *Chorem* structure shown in Chapter 6 to accomplish the mechanism of event and reaction.

Figure 7.9 shows the *Chorem* structure with the additional *Event engine* component.

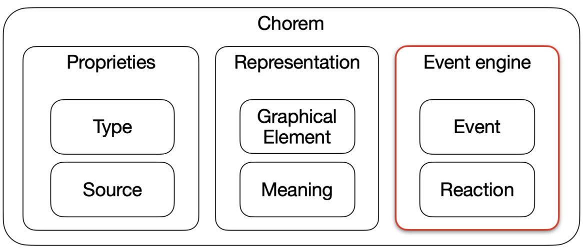



Figure 7.9: The Extended Chorem structure

Compared with the previous structure, the *Iconic Representation* component becomes here simply *Representation*, to remark that the *Graphical Element* can be more than just a static icon. It can be an icon, a graphic animation, or any other graphical element it is possible to display on a screen computer.

The additional *Event engine* component has a set of *Events* triggering correspondent *Reactions*.

The following Figure 7.10 schematizes the possible sources of events to which the *Chorem* can react. It depicts two paths, green and orange.

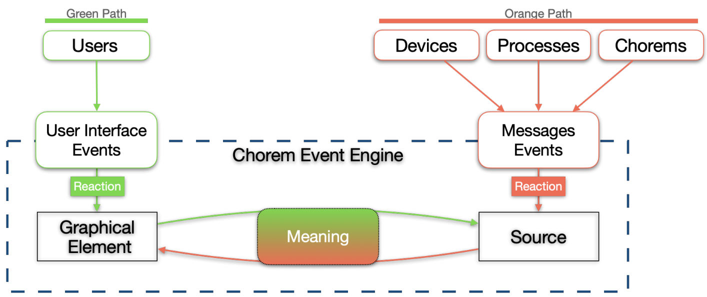



Figure 7.10: Event engine schema

Users can generate GUI events. The *Event Engine* of the destination *Chorem* will perform the appropriate reaction, changing the *Graphical Element*. The change of *Graphical Element* will produce a change in the *Chorem Source* information, keeping the *Meaning* unchanged (green path in Figure 7.10.)

Devices, external processes, and other *Chorems* objects can send messages to a *Chorem*. The *Event Engine* of the destination *Chorem* will perform the changes to the data set that shapes the *Chorem Source* information. Finally, the change of *Chorem Source* will produce a change in the *Graphical Element* to keep the *Meaning* unchanged (orange path in Figure 7.10.)

## 7.4 FRAMEWORK

This section describes a possible framework to deploy these new data visualization tools on Geoportals, Living Labs, and any web portal displaying geographic information.

The framework observes a Client-Server paradigm where the Web browser is the user agent. The following Figure 7.11 depict the proposed framework.

In this framework, the IP *Connected Devices* are sources of event streaming. The *Stream Server* is the server-side solution that collects all these streams in a temporary log. The *Microservices* can

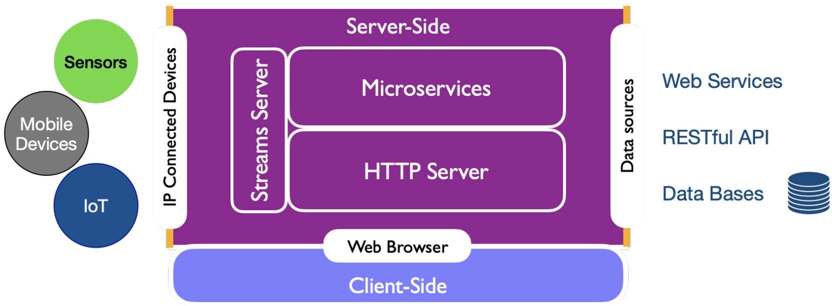



Figure 7.11: Framework

process these logs to send results in real-time to the *HTTP server* or push them to a database for permanent storage.

The *Data Sources* are the traditional sources of data from web services, API, and database connections. *Microservices* can connect to these sources and process the data before sending them to the *HTTP server*. Even the *HTTP server* can connect directly to these sources if no sophisticated data pre-processing is required.

Thus, the HTTP server receives data from all the other components, namely *Stream Server*, *Microservice*, and *Data Source*, serving them to clients through web pages.

## 7.5 ARCHITECTURE

Figure 7.12 depicts a possible layered architecture of the above framework.

The end devices at the lower layer use a modern browser as the user agent.

The server-side elements are deployed on an Edge-Device placed on the LAN.

The geographically distributed data sources are on the Internet resources located in the WAN.

The IP-connected devices ① communicate by Pub/Sub protocols with an MQTT broker ② placed at the server-side.

Microservices ③ can process these data streams and publish new streams with messages of aggregated data or store these data in a local database ④ or distributed one ⑤. Microservices can even process external data sources ⑤ for ETL operations and make these data available to the HTTP Server ⑥.

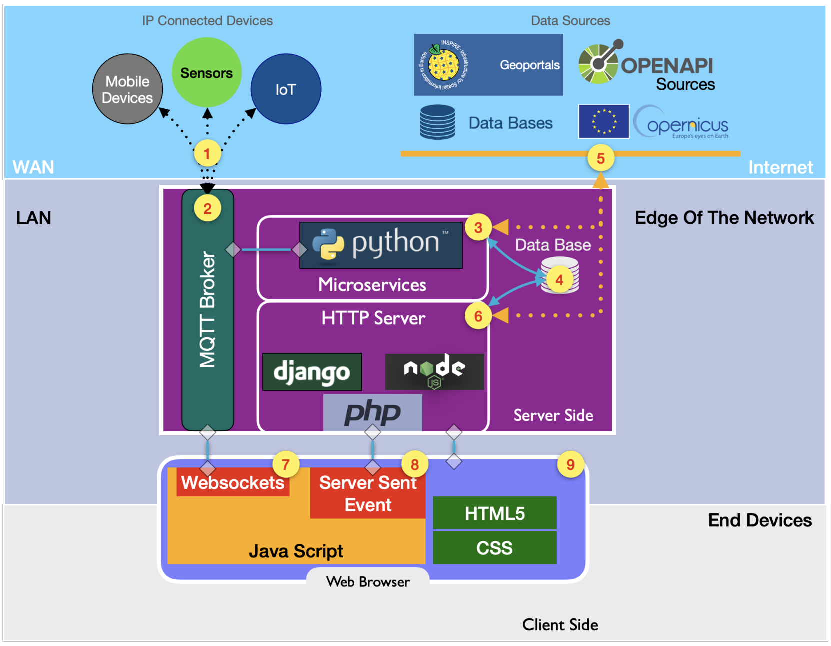



Figure 7.12: Layered architecture for Actionable Chorems

Microservices are Python scripts that communicate with the HTTP server in two ways. Directly, using the Django framework. Indirectly, publishing on the MQTT broker.

The HTTP Server (6) will serve web pages and data. Data can be processed by the Microservices or acquired directly from external data sources through database connectors and Web APIs. PHP scripts obtain data from external sources.

On the Client side, Java Scripts use Websockets (7) to Publish and Subscribe to the MQTT Broker on the server side. By Server-Sent Event protocol (8), the web browser application can receive some events generated on the server-side by PHP scripts.

Finally, standard JavaScript, HTML5, and Cascading Style Sheets (CSS) languages at the Web browser (9) level produce the visualizations implementing the *Chorem* objects.

This architecture allows the realization of web applications and portals able to show data in real-time and from databases.

# 8 CONCLUSIONS

This chapter summarizes the content of the thesis, highlighting the research contribution.

## 8.1 PART I

The studies in the first part of the thesis were related to the emerging solutions to leverage IoT potentiality. The related challenges addressed can be summarized as collecting Big Data, exploiting the broad distribution of analysis processes of this data, and redistributing these results to widely geographically disseminated users.

The focus was on spatiotemporal data and Geographic Information. The results of these studies gave the answers to the first two research questions defined for this research.

*RQ1: How to improve methods and techniques for collecting large volumes of spatiotemporal data when a large network of connected devices is involved?*

The exploration and experimentation of the Edge-Computing paradigm led to the use of Fog-Computing architecture. Its layered structure permitted the distribution of processes to collect and analyze data. These results in a step-by-step data collection and analysis process, avoiding the effort to collect all data to a central point and redistribute the results back to the end-users. Furthermore, since our scope was territorial information, the solution allowed the local process of the data strictly concerning the territory by maintaining the connection with all the other widely disseminated pieces of information.

*RQ2 - How to process this data when a real-time awareness is required?*

The question was particularly relevant when considering real-time data that need to achieve a real-time decision. The Fog-computing architecture effectively reduced the latency, bringing the computation closer to the data producers and consumers. Furthermore, in many cases, avoiding the Internet connection results in a more reliable and inexpensive network connection in the local area.

## 8.2 PART II

The second part of the thesis investigates data exploration and information dissemination using web technologies. The goal was to propose the enrichment of the geovisual analytical toolset with a new one that can be used by non-experts and can also include data from IoT sources.

In these three years as a doctoral student, especially at conferences, I have seen that many tools initially designed for citizens or for particular productive sectors such as farmers have failed in their objectives. In fact, often these, while well used by the experts were not then used by the users they were intentionally aimed at. The main reason turned out to be the difficulty of use by people inexperienced in data analysis and computer media, such as precisely most citizens and farmers.

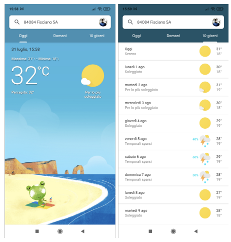
It had been observed that even if they made use of visual communication, which is known to be more effective than textual communication, they still did not achieve the goal.

*RQ3 - How improve the geovisual analysis tools for BDSD?*

An attempt was made providing a tool from which it is possible to ask for answers without necessarily having to know how to get them. This criterion is especially important when the user has no experience or training in the techniques adopted in the domain, but knows only the questions to which he or she wants answers.

To clarify the concept an example is depicted below. Figure 8.1a shows a screenshot of a weather forecast program that was broadcast on television in 1968. To answer the simple question: will it rain tomorrow? A weatherman showed a weather map, trying to explain to the television audience what all those lines and numbers meant. In contrast, Figure 8.1b shows a modern and popular application that answers the question in a very simple and straightforward way.

(a) TV weather forecast in 1968




(b) Weather forecast Application

Thus, in Chapter 7 the *Chorematic Display* composed of actionable *Chorems* was proposed, with the aim of making visual analysis more intuitive and simple, without requiring the user to have in-depth knowledge of the data source or its metadata, thus adding a new tool to the geovisual analytical toolbox.

# 9 FUTURE WORK

## CAN MACHINE LEARNING BE INTEGRATED WITH THE PROPOSED LAYERED ARCHITECTURES?

In [10–12], we experimented with AI at the edge of the network. Mist-Fog architectures were designed to reduce latency by performing inference on edge devices and the local network.

These experiences have increased the interest in the opportunity provided by modern Edge-Devices to perform inference Machine Learning (ML) avoiding cloud resources and the consequent Internet connection, which in many cases is expensive and unreliable. Actually more than 100 companies are proposing ML Inference dedicate chips, ranging from Mobile-Devices to Edge-Devices up to HPC [107].

In the context of BDSD, an important role is played by modern Earth Observation (EO) resources, which most often consist of satellite images. Here, AI is intensively used to extract data from images and is mostly performed on HPC, so bringing these AI tasks to the Edge is still promising and remains a future work.

## WHAT ROLE CAN SUPERVISED ALGORITHMS PLAY IN THIS CONTEXT?

Supervised learning is defined by the use of labeled datasets to train algorithms that can classify data or accurately predict outcomes. In the context of EO and in particular in remote sensing, this AI training activity implies the use of huge amount of labeled images. These images sources are usually on Cloud, because periodically produced by satellite constellation owned by big consortium of companies, labeled and validate by on field data, and worldwide available.

Therefore, bringing the training of this AI to edge devices requires the transfer of huge data sets from the cloud to the edge of the network, a large amount of memory resources to store them, and a powerful computing resource to train the AI model. Consequently, the time and cost required to transfer training datasets could invalidate the low latency advantage claimed by the Edge architecture. Moreover, large storage resources and powerful hardware are not typical features of edge devices, but mainly of cloud data centers, which can be considered virtually unlimited in terms of resources.

Hence, the investigation of how to perform training in a distributed and layered architecture, such as hybrid Cloud-Fog-Computing, is still promising and is already emerging in the literature [139] giving rise to the so-called Edge-Intelligence (EI).

## WHAT ROLE CAN UNSUPERVISED ALGORITHMS PLAY IN THIS CONTEXT?

Unlike supervised learning, unsupervised learning uses unlabeled data. It saves the time required to label training sets, but retains the problem of the amount of training datasets consisting of satellite images. In the context of EO the unsupervised learning has been experimented by the authors of [121] to estimate the soil moisture from remote sensing images. The results were not satisfactory, compared with a supervised solution which uses labeled data validated by input of field data.

By other end unsupervised learning can discover patterns that help solve clustering or association problems. This is especially useful when users are unsure of the common properties of a data set. Then, assuming that the data are produced at the Edge and that the results are also consumed in the same territory where they are produced, we see the possibility of using unsupervised training on Edge-Device as well, since the path that the data have to take is small and probably the whole thing is done on the local network.

In [8, 13, 15] we hypothesized such a solution for a domain other than territorial awareness. In these works, a novel tactile and tangible interface, the TactCube, allows the user to interact with an Ambient Intelligence (AmI). Ambient Intelligence is a vision of daily life in which smart devices interact with humans to make their lives easier and technology is invisible. AI governs this smart ambient and interacts with humans to best meet their needs.

The TactCube is designed as a Mist-Node of a Mist-Fog architecture. It transmits manipulations made by the user to the AI that governs the environment, that is designed as a Fog-Node. The Fog-Node offers computational and connectivity resources to others Fog-Nodes and Mist-Nodes in the local area network. It is connected to Internet to collects news and information that could contextualize better the user requests. The main task of AI is to interpret the user's TactCube manipulations as conversational messages. Since the TactCube provides multiple profiles that can be used by different users, training does not occur once as in previous cases, but more frequently. In addition, the interface manipulations, which represent the data to be processed, and the responses of the AmI, which represent the output of the AI, reside in the same local network. All these considerations justify the use of EI.

## WHAT ROLE CAN REINFORCEMENT LEARNING ALGORITHMS PLAY IN THIS CONTEXT?

Usually a ML has two phases, the training and the inference. Placing the training activities in the Cloud and the Inference at the Edge is the most popular solution. The main reasons are that the training is preferable to Could because it is required only once, or repeated only when the nature of the input data changes. Moreover, It requires large memory and computational resources, which are often readily available in the cloud. Instead, the Inference, which it is applied every time the AI elaborates an answer, can be deployed at the Edge, since the emerging of dedicated chips and edge-devices, and the modest storage resource required.

In the case of a Reinforcement Learning (RL) algorithm, the previous two steps are combined , and the system learns by working. An agent evaluates the results, rewarding the solutions closest to the desired behavior and punishing the less desired ones. In this way, step by step the intended solutions emerge and the wrong ones automatically disappear.

An interesting use of RL is given by [95]. The authors proposed a solution based on RL to the Computation Offloading problem in a Mobile Edge Computing (MobEC). As future work, it might be interesting to compare the results in terms of energy savings and latency of this solution with the one proposed in Chapter 3, based instead on a *Collaborate and Compete* approach.

Another interesting investigation of the RL approach could be made in the creation of AI behind the interpretation of the TactCube manipulation. The users could teach the AI how to interpret their manipulations directly using the interface, without a prior training phase.

### CAN MACHINE LEARNING CONCEPTS SUCH AS CLASSIFICATION AND CLUSTERING PLAY A ROLE IN IMPROVING THE PROPOSED ARCHITECTURE? OR CAN THEY ONLY BE USEFUL IN ANALYZING THE DATA GENERATED BY THE ARCHITECTURE?

The concept of machine learning is widely used in remote sensing, bringing undoubted advantages in extracting information from sensor data and satellite images. [25, 65, 102, 113] are some of many examples present in literature. It is also easy to find many works in the literature on ML as a solution for anomaly detection and imputation of missing data in IoT solutions. Examples include: [40, 119, 126]. Therefore, how AI can be useful for data analysis is widely studied with appreciable results .

In addition, ML can also play a role in improving the Fog-Computing architecture. Since the Fog-Computing architecture relies on efficient workload distribution and reliable network communication, it is promising to study ML solutions to optimize these features. In particular, the opportunity to overcome the challenge of training ML at the Edge by appropriately distributing the training of ML on nodes for parallel processing is of interest. Some theoretical examples can be found in the literature, [122, 129], but more experimentation and a pragmatic approach are promising for future work.

# BIBLIOGRAPHY


[1] Mahbuba Afrin, Jiong Jin, Akhlaqur Rahman, Ashfaqur Rahman, Jiafu Wan, and Ekram Hossain. "Resource Allocation and Service Provisioning in Multi-Agent Cloud Robotics: A Comprehensive Survey." In: *CoRR* abs/2104.14270 (2021).

[2] Analog Devices. *3-Axis Digital Accelerometer*. URL: https://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf (visited on 11/30/2020).

[3] Geraldine André et al. "LambdAgrIoT: a new architecture for agricultural autonomous robots'scheduling: from design to experiments." In: *Cluster Computing* (2022). DOI: 10.1007/s10586-022-03592-5. URL: https://doi.org/10.1007/s10586-022-03592-5.

[4] Louis Arreghini. "La modélisation graphique dans la réalisation des atlas pour le développement." In: *Les atlas pour le développement en coopération* (1995), pp. 1–10.

[5] B Bachelet, P Battistoni, S Bimonte, C Cariou, G Chalhoub, F Coutarel, and N Tricot. "Towards an architecture for online scheduling of autonomous robots in agriculture: open issues." In: *International Journal of Smart Vehicles and Smart Transportation* (2021).

[6] Christophe Batardy. "Le Berry antique-De la carte au modèle-chorème." In: *Revue archéologique du Centre de la France* 43 (2005), pp. 253–258.

[7] P Battistoni, M Grimaldi, M Sebillo, and G Vitiello. "Living Labs and Open Innovation to support local development policies." In: 2021.

[8] Pietro Battistoni, Marianna Di Gregorio, Marco Romano, Monica Sebillo, and Giuliana Vitiello. "TactCube: An Intelligent Device to converse with Smart Environments." In: *Sensors* 22.14 (2022). ISSN: 1424-8220. DOI: 10.3390/

s22145235. URL: https://www.mdpi.com/1424-8220/22/14/5235.

[9] Pietro Battistoni, Marianna Di Gregorio, Domenico Giordano, Monica Sebillo, Genoveffa Tortora, and Giuliana Vitiello. "Wearable Interfaces and Advanced Sensors to Enhance Firefighters Safety in Forest Fires." In: Association for Computing Machinery, 2020. ISBN: 9781450375351. DOI: 10.1145/3399715.3399961. URL: https://doi.org/10.1145/3399715.3399961.

[10] Pietro Battistoni, Marianna Di Gregorio, Marco Romano, Monica Sebillo, and Giuliana Vitiello. "AI at the Edge for Sign Language Learning Support." In: *Humanized Computing and Communication* Vol. 1 (No. 1 2020), pp. 23–42. DOI: 10.35708/hcc1868-148363.

[11] Pietro Battistoni, Marianna Di Gregorio, Marco Romano, Monica Sebillo, Giuliana Vitiello, and Giandomenico Solimando. "Sign Language Interactive Learning - Measuring the User Engagement." In: ed. by Panayiotis Zaphiris and Andri Ioannou. Springer International Publishing, 2020, pp. 3–12. ISBN: 978-3-030-50506-6. DOI: 10.1007/978-3-030-50506-6_1.

[12] Pietro Battistoni, Marianna Di Gregorio, Monica Sebillo, and Giuliana Vitiello. "AI at the Edge for Sign Language Learning Support." In: IEEE, Sept. 2019. DOI: 10.1109/hcc46620.2019.00011. URL: https://doi.org/10.1109%2Fhcc46620.2019.00011.

[13] Pietro Battistoni, Marianna di Gregorio, Marco Romano, Monica Sebillo, and Giuliana Vitiello. "TactCube: Designing Mobile Interactions with Ambient Intelligence." In: ed. by Carmelo Ardito, Rosa Lanzilotti, Alessio Malizia, Helen Petrie, Antonio Piccinno, Giuseppe Desolda, and Kori Inkpen. Springer International Publishing, 2021, pp. 599–609. ISBN: 978-3-030-85610-6. DOI: 10.1007/978-3-030-85610-6_34.

[14] Pietro Battistoni, Michele Grimaldi, Marco Romano, Monica Sebillo, and Giuliana Vitiello. "Interactive Maps of Chorems Explaining Urban Contexts to Align Smart Community's Actors." In: ed. by Osvaldo Gervasi, Beniamino

Murgante, Sanjay Misra, Chiara Garau, Ivan Blečić, David Taniar, Bernady O Apduhan, Ana Maria A C Rocha, Eufemia Tarantino, and Carmelo Maria Torre. Springer International Publishing, 2021, pp. 549–564. ISBN: 978-3-030-86976-2. DOI: 10.1007/978-3-030-86976-2_37.

[15] Pietro Battistoni and Monica Sebillo. "A Tactile User Device to Interact with Smart Environments." In: ed. by Helmut Degen and Stavroula Ntoa. Springer International Publishing, 2021, pp. 461–471. ISBN: 978-3-030-77772-2. DOI: 10.1007/978-3-030-77772-2_30.

[16] Pietro Battistoni, Monica Sebillo, and Giuliana Vitiello. "Computation Offloading with MQTT Protocol on a Fog-Mist Computing Framework." In: ed. by Raffaele Montella, Angelo Ciaramella, Giancarlo Fortino, Antonio Guerrieri, and Antonio Liotta. Springer International Publishing, 2019, pp. 140–147. ISBN: 978-3-030-34914-1. DOI: 10.1007/978-3-030-34914-1_14.

[17] Pietro Battistoni, Monica Sebillo, and Giuliana Vitiello. "Experimenting with a Fog-computing Architecture for Indoor Navigation." In: 2019, pp. 161–165. DOI: 10.1109/FMEC.2019.8795307.

[18] Pietro Battistoni, Monica Sebillo, and Giuliana Vitiello. "An IoT-Based Mobile System for Safety Monitoring of Lone Workers." In: *IoT* 2 (3 2021), pp. 476–497. ISSN: 2624-831X. DOI: 10.3390/iot2030024. URL: https://www.mdpi.com/2624-831X/2/3/24.

[19] A Belhassena et al. "Towards an Architecture for Agricultural Autonomous Robots' Scheduling." In: Oct. 2021.

[20] Y. Bertacchini, Marisela Rodriguez-Salvador, and Wahida Souari. "From territorial intelligence to compositive & sustainable system. Case studies in Mexico & in Gafsa university." In: *In International Conference of Territorial Intelligence, Huelva 2007*. Huelva, Spain, Oct. 2007, p. 106–124. URL: https://halshs.archives-ouvertes.fr/halshs-00516033.

[21] Bluetooth SIG. *Alert Notification Service*. URL: https://www.bluetooth.com/specifications/gatt/ (visited on 11/24/2020).

[22] Bluetooth SIG. *Core Specification*. URL: https://www.bluetooth.com/specifications/bluetooth-core-specification/ (visited on 11/24/2020).

[23] Roger Brunet. "La carte-modèle et les chorèmes." In: *Mappemonde* (1986).

[24] G. Calice, A. Mtibaa, R. Beraldi, and H. Alnuweiri. "Mobile-to-mobile opportunistic task splitting and offloading." In: *2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. Oct. 2015, pp. 565–572. DOI: 10.1109/WiMOB.2015.7348012.

[25] Manuel Campos-Taberner, Francisco Javier García-Haro, Beatriz Martínez, Emma Izquierdo-Verdiguier, Clement Atzberger, Gustau Camps-Valls, and María Amparo Gilabert. "Understanding deep learning in land use classification based on Sentinel-2 time series." In: *Scientific reports* 10.1 (2020), pp. 1–12.

[26] Michael Parmisano Canares, Dave Marcial, and Marijoe Narca. "Enhancing Citizen Engagement with Open Government Data." In: *The Journal of Community Informatics* 12.2 (June 2016). DOI: 10.15353/joci.v12i2.3240. URL: https://doi.org/10.15353\%2Fjoci.v12i2.3240.

[27] Maurizio Capra, Riccardo Peloso, Guido Masera, Massimo Ruo Roch, and Maurizio Martina. "Edge Computing: A Survey On the Hardware Requirements in the Internet of Things World." In: *Future Internet* 11.4 (2019). ISSN: 1999-5903. DOI: 10.3390/fi11040100. URL: https://www.mdpi.com/1999-5903/11/4/100.

[28] Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman. *Readings in information visualization - using vision to think*. Academic Press, 1999.

[29] Christophe Cariou, Zoltan Gobor, Benjamin Seiferth, and Michel Berducat. "Mobile Robot Trajectory Planning Under Kinematic and Dynamic Constraints for Partial and Full Field Coverage." In: *Journal of Field Robotics* 34.7

(May 2017), pp. 1297–1312. DOI: 10.1002/rob.21707. URL: https://doi.org/10.1002%2Frob.21707.

[30] Chunhee Cho and JeeWoong Park. "An Embedded Sensory System for Worker Safety: Prototype Development and Evaluation." In: *Sensors* 18.4 (2018). ISSN: 1424-8220. DOI: 10.3390/s18041200. URL: https://www.mdpi.com/1424-8220/18/4/1200.

[31] Cisco. *Cisco Annual Internet Report (2018–2023) White Paper*. URL: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html (visited on 12/16/2020).

[32] Davide De Chiara, Vincenzo Del Fatto, Robert Laurini, Monica Sebillo, and Giuliana Vitiello. "A chorem-based approach for visually analyzing spatial data." In: *Journal of Visual Languages & Computing* 22.3 (2011), pp. 173–193.

[33] Pasquale De Muro, Matteo Mazziotta, and Adriano Pareto. "Composite Indices of Development and Poverty: An Application to MDGs." In: *Social Indicators Research* 104.1 (2011), pp. 1–18.

[34] Olivier Debauche, Meryem El Moulat, Saïd Mahmoudi, Pierre Manneback, and Frédéric Lebeau. "Irrigation pivot-center connected at low cost for the reduction of crop water requirements." In: *Int. Conf. on Advanced Communication Technologies and Networking (CommNet)*. IEEE. 2018, pp. 1–9.

[35] Vincenzo Del Fatto, Robert Laurini, Karla Lopez, Rosalva Loreto, Françoise Milleret-Raffort, Monica Sebillo, David Sol-Martinez, and Giuliana Vitiello. "Potentialities of Chorems as Visual Summaries of Geographic Databases Contents." In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2007, pp. 537–548.

[36] Vincenzo Del Fatto, Robert Laurini, Karla Lopez, Monica Sebillo, and Giuliana Vitiello. "A chorem-based approach for visually synthesizing complex phenomena." In: *Information Visualization* 7.3-4 (2008), pp. 253–264.

[37] Mathieu Deremetz., Adrian Couvent., Roland Lenain., Benoit Thuilot., and Christophe Cariou. "A Generic Control Framework for Mobile Robots Edge Following." In: *Proc. of Int. Conf. on Informatics in Control, Automation and Robotics*. 2019, pp. 104–113.

[38] Vincenzo Deufemia, Massimiliano Giordano, Giuseppe Polese, and Genoveffa Tortora. "A visual language-based system for extraction–transformation–loading development." In: *Software: Practice and Experience* 44.12 (2014), pp. 1417–1440. DOI: https://doi.org/10.1002/spe.2201. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/spe.2201. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.2201.

[39] Google Developers. *Get Started with Beacons*. Tech. rep. Google Developers, 2021.

[40] Thomas G. Dietterich and Tadesse Zemicheal. *Anomaly Detection in the Presence of Missing Values*. 2018. DOI: 10.48550/ARXIV.1809.01605. URL: https://arxiv.org/abs/1809.01605.

[41] Philippe Dobbelaere and Kyumars Sheykh Esmaili. "Kafka versus RabbitMQ: A comparative study of two industry reference publish/subscribe implementations: Industry Paper." In: *ACM Int. Conf. on Distributed and Event-based Systems (DEBS)*. 2017, pp. 227–238.

[42] *ESP8266 Non-OS SDK API References*. July 2021. URL: https://www.espressif.com/sites/default/files/documentation/2c-esp8266_non_os_sdk_api_reference_en.pdf (visited on 07/18/2021).

[43] *ESPlorer*. URL: https://esp8266.ru/esplorer/ (visited on 07/18/2021).

[44] Cheikh Kacfah Emani, Nadine Cullot, and Christophe Nicolle. "Understandable Big Data: A survey." In: *Computer Science Review* 17 (2015), pp. 70–81.

[45] Luis Emmi, Mariano Gonzalez-de Soto, Gonzalo Pajares, and Pablo Gonzalez-de Santos. "New trends in robotics for agriculture: integration and assessment of a real fleet of robots." In: *The Scientific World J.* 2014 (2014).

[46] Eric Gakstatter. *What Exactly Is GPS NMEA Data?* URL: https://www.gpsworld.com/what-exactly-is-gps-nmea-data/ (visited on 12/02/2020).

[47] Espressif. *A feature-rich MCU with integrated Wi-Fi and Bluetooth.* URL: https://www.espressif.com/en/products/socs/esp32 (visited on 11/30/2020).

[48] EspressifSystems. *ESP32 Technical Reference Manual*. Tech. rep. Espressif Systems, 2021.

[49] Patrick Th. Eugster, Pascal A. Felber, Rachid Guerraoui, and Anne-Marie Kermarrec. "The Many Faces of Publish/Subscribe." In: *ACM Comput. Surv.* 35.2 (2003), pp. 114–131. ISSN: 0360-0300. DOI: 10.1145/857076.857078. URL: https://doi.org/10.1145/857076.857078.

[50] Euopean Network of Living Labs. Website. URL: https://enoll.org/.

[51] European Agency for Safety and Healt at Work. *Directive 89/656/EEC - use of personal protective equipment*. URL: https://osha.europa.eu/en/legislation/directives/4 (visited on 12/14/2020).

[52] European Agency for Safety and Health at Work. *Smart Personal Protective Equipment: Intelligent Protection for the Future*. URL: https://osha.europa.eu/en/publications/smart-personal-protective-equipment-intelligent-protection-future/view (visited on 05/25/2021).

[53] European Environment Agency. *Explore air pollution data*. 2022. URL: https://www.eea.europa.eu/themes/air/explore-air-pollution-data (visited on 03/08/2022).

[54] European Environment Information and Observation Network. *European Air Quality*. URL: https://www.eea.europa.eu/themes/air/air-quality-index/index.

[55] European Union's Occupational Safety and Health. *OSH in general*. URL: https://oshwiki.eu/wiki/OSH_in_general#Costs_of_work_accidents_and_occupational_diseases (visited on 12/14/2020).

[56] L. Faramondi, P. Bragatto, C. Fioravanti, M. G. Gnoni, S. Guarino, and R. Setola. “A Wearable Platform to Identify Workers Unsafety Situations.” In: *2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4.0 IoT)*. 2019, pp. 339–343. DOI: 10.1109/METROI4.2019.8792857.

[57] Vincenzo Del Fatto, Robert Laurini, Karla Lopez, Monica Sebillo, and Giuliana Vitiello. “A Chorem based Method for Visualizing Evolutionary Scenarios.” In: *Information Visualization journal*. 2008.

[58] Patrice Flichy. “Connected Individualism between Digital Technology and Society.” In: *Réseaux* 124 (2004), pp. 17–51.

[59] Eclipse Fundation. *paho - Python Client documentation*. Tech. rep. Eclipse Fundation, 2021.

[60] Tera Marie Green and Ross Maciejewski. “A Role for Reasoning in Visual Analytics.” In: *2013 46th Hawaii International Conference on System Sciences*. 2013, pp. 1495–1504. DOI: 10.1109/HICSS.2013.58.

[61] M. A. M. de Groot-Reichwein, R. J. A. van Lammeren, H. Goosen, A. Koekoek, A. K. Bregt, and P. Vellinga. “Urban heat indicator map for climate adaptation planning.” In: *Mitigation and Adaptation Strategies for Global Change* 23.2 (2018), pp. 169–185.

[62] OpenFog Consortium Architecture Working Group. *OpenFog Reference Architecture for Fog Computing*. Tech. rep. OpenFog Consortium, 2017.

[63] Guenter Hesse, Christoph Matthies, and Matthias Uflacker. “How Fast Can We Insert? An Empirical Performance Evaluation of Apache Kafka.” In: *IEEE Int. Conf. on Parallel and Distributed Systems (ICPADS)*. 2020, pp. 641–648.

[64] ISTAT. *Web service SDMX per l’accesso ai dati di IStat*. Website. Oct. 2020. URL: https://www.istat.it/it/metodi-e-strumenti/web-service-sdmx.

[65] Dino Ienco, Roberto Interdonato, Raffaele Gaetano, and Dinh Ho Tong Minh. "Combining Sentinel-1 and Sentinel-2 Satellite Image Time Series for land cover mapping via a multi-source deep learning architecture." In: *ISPRS Journal of Photogrammetry and Remote Sensing* 158 (2019), pp. 11–22.

[66] Nadeem Iftikhar, Bartosz Piotr Lachowicz, Akos Madarasz, Finn Ebertsen Nordbjerg, Thorkil Baattrup-Andersen, and Karsten Jeppesen. "Real-time Visualization of Sensor Data in Smart Manufacturing using Lambda Architecture." In: *Int. Conf. on Data Science, Technology and Applications (DATA)*. 2020, pp. 215–222.

[67] Internet Engineering Task Force. *The WebSocket Protocol*. URL: https://tools.ietf.org/html/rfc6455 (visited on 11/30/2020).

[68] Michaela Iorga, Larry Feldman, Robert Barton, Michael J Martin, Ned Goren, and Charif Mahmoudi. *Fog computing conceptual model*. Tech. rep. Mar. 2018. DOI: 10.6028/nist.sp.500-325. URL: https://doi.org/10.6028%2Fnist.sp.500-325.

[69] V. Jayasree and M. N. Kumari. "IOT Based Smart Helmet for Construction Workers." In: *2020 7th International Conference on Smart Structures and Systems (ICSSS)*. 2020, pp. 1–5. DOI: 10.1109/ICSSS49621.2020.9202138.

[70] M. Jeong, H. Lee, M. Bae, D. Shin, S. Lim, and K. B. Lee. "Development and Application of the Smart Helmet for Disaster and Safety." In: *2018 International Conference on Information and Communication Technology Convergence (ICTC)*. 2018, pp. 1084–1089. DOI: 10.1109/ICTC.2018.8539625.

[71] Ning Jia. *Detecting Human Falls with a 3-Axis Digital Accelerometer*. URL: https://www.analog.com/en/analog-dialogue/articles/detecting-falls-3-axis-digital-accelerometer.html (visited on 12/10/2020).

[72] R. Kanan, O. Elhassan, R. Bensalem, and A. Husein. "A wireless safety detection sensor system." In: *2016 IEEE*

*SENSORS*. 2016, pp. 1–3. DOI: 10.1109/ICSENS.2016.7808926.

[73] Daniel A Keim, Florian Mansmann, Jörn Schneidewind, and Hartmut Ziegler. "Challenges in visual data analysis." In: *Tenth International Conference on Information Visualisation (IV'06)*. IEEE. 2006, pp. 9–16.

[74] Kimberly-Clark Professional. *Alarming Number of Workers Fail to Wear Required Protective Equipment*. URL: https://investor.kimberly-clark.com/news-releases/news-release-details/alarming-number-workers-fail-wear-required-protective-equipment (visited on 12/14/2020).

[75] Robert Laurini. *Geographic Knowledge Infrastructure. Applications to Territorial Intelligence and Smart Cities*. Elsevier Ltd, 2017.

[76] Robert Laurini, Monica Sebillo, Giuliana Vitiello, David Sol-Martinez, Françoise Raffort, Sébastien Gadal, and Françoise Raffort. *Computer-generated Visual Summaries of Spatial Databases: Chorems or not Chorems?* May 2009. URL: http://journals.openedition.org/sapiens/795.

[77] Minh Le, Stephen Clyde, and Young-Woo Kwon. "Enabling multi-hop remote method invocation in device-to-device networks." In: *Human-centric Computing and Information Sciences* 9.1 (June 2019), p. 20. ISSN: 2192-1962. DOI: 10.1186/s13673-019-0182-9. URL: https://doi.org/10.1186/s13673-019-0182-9.

[78] Minh Le and Young-Woo Kwon. "Utilizing Nearby Computing Resources for Resource-limited Mobile Devices." In: *Proceedings of the Symposium on Applied Computing*. SAC '17. New York, NY, USA: ACM, 2017, pp. 572–575. ISBN: 978-1-4503-4486-9. DOI: 10.1145/3019612.3019884. URL: http://doi.acm.org/10.1145/3019612.3019884.

[79] Shinho Lee, Hyeonwoo Kim, Dong-kweon Hong, and Hongtaek Ju. "Correlation analysis of MQTT loss and delay according to QoS level." In: *The International Conference on Information Networking 2013 (ICOIN)*. 2013, pp. 714–717. DOI: 10.1109/ICOIN.2013.6496715.

[80] H. Li. “Research on safety monitoring system of workers in dangerous operation area of port.” In: *2017 4th International Conference on Transportation Information and Safety (ICTIS)*. 2017, pp. 400–408. DOI: 10.1109/ICTIS.2017.8047796.

[81] Roger A. Light. “Mosquitto: server and client implementation of the MQTT protocol.” In: *Journal of Open Source Software* 2.13 (2017), p. 265. DOI: 10.21105/joss.00265. URL: https://doi.org/10.21105/joss.00265.

[82] Aristidis Likas, Nikos Vlassis, and Jakob J. Verbeek. “The global k-means clustering algorithm.” In: *Pattern Recognition* 36.2 (2003), pp. 451–461.

[83] W. Luo and Q. Wang. “Hardhat-Wearing Detection with Cloud-Edge Collaboration in Power Internet-of-Things.” In: *2019 4th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*. 2019, pp. 681–6813. DOI: 10.1109/ICMCCE48743.2019.00158.

[84] Alan M. MacEachren, Mark Gahegan, William Pike, Isaac Brewer, Guoray Cai, Eugene Lengerich, and Frank Hardisty. “Geovisualization for Knowledge Construction and Decision Support.” In: *IEEE Computer Graphics and Applications* 24.1 (2004), pp. 13–17.

[85] Nathan Marz and James Warren. *Big Data: Principles and best practices of scalable realtime data systems*. Manning, 2015.

[86] Matteo Mazziotta and Adriano Pareto. “On a Generalized Non-compensatory Composite Index for Measuring Socio-economic Phenomena.” In: *Social Indicators Research* 127.3 (2016), pp. 983–1003.

[87] K. M. Mehata, S. K. Shankar, N. Karthikeyan, K. Nandhinee, and P. R. Hedwig. “IoT Based Safety and Health Monitoring for Construction Workers.” In: *2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)*. 2019, pp. 1–7. DOI: 10.1109/ICIICT1.2019.8741478.

[88] Engineering at Meta. *Building Facebook Messenger*. Tech. rep. Meta, 2011.

[89] E Morozov and F Bria. "Rethinking the Smart City. Democratizing Urban Technology, in "Rosa Luxemburg Stiftung"." In: *City Series,(5)* (2018).

[90] United Nations. *Big Data for Sustainable Development*. URL: https://www.un.org/en/global-issues/big-data-for-sustainable-development (visited on 03/13/2022).

[91] Emerson Navarro, Nuno Costa, and António Pereira. "A Systematic Review of IoT Solutions for Smart Farming." In: *Sensors* 20.15 (2020), p. 4231.

[92] BARACK OBAMA. *Transparency and Open Government*. online. Jan. 2009.

[93] Barack Obama. "Transparency and open government: memorandum for the heads of executive departments and agencies. 2009." In: *Retrieved September* 6 (2009), p. 2016.

[94] *Overview - NodeMCU Documentation*. Feb. 2021. URL: https://nodemcu.readthedocs.io/en/dev-esp32/ (visited on 07/18/2021).

[95] Shengli Pan, Zhiyong Zhang, Zongwang Zhang, and Deze Zeng. "Dependency-Aware Computation Offloading in Mobile Edge Computing: A Reinforcement Learning Approach." In: *IEEE Access* 7 (2019), pp. 134742–134753. DOI: 10.1109/ACCESS.2019.2942052.

[96] Arjun Pandya, Oluwatobiloba Odunsi, Chen Liu, Alfredo Cuzzocrea, and Jianwu Wang. "Adaptive and Efficient Streaming Time Series Forecasting with Lambda Architecture and Spark." In: *IEEE Int. Conf. on Big Data*. IEEE, 2020, pp. 5182–5190.

[97] M. Park, S. Park, M. Song, and S. Park. "IoT-based Safety Recognition Service for Construction Site." In: *2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN)*. 2019, pp. 738–741. DOI: 10.1109/ICUFN.2019.8806080.

[98] Open Governament Partnership. *Open Government Declaration*. online. Sept. 2011.

[99] Jay Patel. "Introduction to Web Scraping." In: Apress Berkeley CA, Nov. 2020, pp. 1–30. ISBN: 978-1-4842-6575-8. DOI: 10.1007/978-1-4842-6576-5_1.

[100] Caron Patrick. "Modélisation graphique et chorèmes: la gestion des parcours collectifs à Massaroca (Brésil du Nordeste)." In: *Mappemonde* 62 (June 2001).

[101] Maria Angela Pellegrino, Luca Postiglione, and Vittorio Scarano. "Detecting Data Accuracy Issues in Textual Geographical Data by a Clustering-Based Approach." In: *8th ACM IKDD CODS and 26th COMAD*. CODS COMAD 2021. New York, NY, USA: Association for Computing Machinery, 2021, pp. 208–212. ISBN: 9781450388177. DOI: 10.1145/3430984.3431031. URL: https://doi.org/10.1145/3430984.3431031.

[102] Charlotte Pelletier, Geoffrey I Webb, and François Petitjean. "Deep learning for the classification of Sentinel-2 image time series." In: *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2019, pp. 461–464.

[103] George Percivall. "Realizing the geospatial potential of mobile, IoT and big data." In: *Int. Conf. on Computing for Geospatial Research and Application*. Ed. by Lindi Liao. ACM, 2012, p. 8.

[104] UN Global Pulse. "BIG DATA FOR DEVELOPMENT: A PRIMER." In: *Global Pulse* (2013). URL: https://www.un.org/en/global-issues/big-data-for-sustainable-development.

[105] Quectel. *GNSS Module with Chip Antenna*. URL: https://www.quectel.com/product/l96.htm (visited on 11/30/2020).

[106] Ahmed Memon RIZWAN, Leung Y.C. DENNIS, and Chunho LIU. "A review on the generation, determination and mitigation of Urban Heat Island." In: *Journal of Environmental Sciences* 20.1 (2008), pp. 120–128.

[107] Vijay Janapa Reddi et al. *MLPerf Inference Benchmark*. 2019. DOI: 10.48550/ARXIV.1911.02549. URL: https://arxiv.org/abs/1911.02549.

[108] Ignacio Rodríguez Garzón, Myriam Martínez Fiestas, Antonio Delgado Padial, and Valeriano Lucas Ruiz. "Perception of Occupational Risk of Firefighters in Quito (Ecuador)." In: *Fire Technology* 52.3 (2016), pp. 753–773.

DOI: 10.1007/s10694-015-0494-x. URL: https://doi.org/10.1007/s10694-015-0494-x.

[109] Marco Romano, Paloma Díaz, and Ignacio Aedo. "Emergency Management and Smart Cities: Civic Engagement Through Gamification." In: *Information Systems for Crisis Response and Management in Mediterranean Countries*. Ed. by Paloma Díaz, Narjès Bellamine Ben Saoud, Julie Dugdale, and Chihab Hanachi. Cham: Springer International Publishing, 2016, pp. 3–14.

[110] Monica Maria Lucia Sebillo, Giuliana Vitiello, Michele Grimaldi, and Davire De Chiara. "A Territorial Intelligence-based Approach for Smart Emergency Planning." In: vol. In Data Science and Big Data Analytics in Smart Environments, M. Chinnici. Routledge Taylor & Francis Group, 2021, pp. 135–156.

[111] Monica Sebillo, Genny Tortora, Maurizio Tucci, Giuliana Vitiello, Athula Ginige, and Pasquale Di Giovanni. "Combining personal diaries with territorial intelligence to empower diabetic patients." In: *Journal of Visual Languages & Computing* 29 (Aug. 2015), pp. 1–14.

[112] S. Shabina. "Smart Helmet Using RF and WSN Technology for Underground Mines Safety." In: *2014 International Conference on Intelligent Computing Applications*. 2014, pp. 305–309. DOI: 10.1109/ICICA.2014.105.

[113] Yuri Shendryk, Yannik Rist, Catherine Ticehurst, and Peter Thorburn. "Deep learning for multi-modal classification of cloud, shadow and land cover scenes in PlanetScope and Sentinel-2 imagery." In: *ISPRS Journal of Photogrammetry and Remote Sensing* 157 (2019), pp. 124–136.

[114] Weisong Shi, George Pallis, and Zhiwei Xu. "Edge Computing [Scanning the Issue]." In: *Proceedings of the IEEE* 107.8 (2019), pp. 1474–1481. DOI: 10.1109/JPROC.2019.2928287.

[115] B. Shneiderman. "The eyes have it: a task by data type taxonomy for information visualizations." In: *Proceedings 1996 IEEE Symposium on Visual Languages*. 1996, pp. 336–343. DOI: 10.1109/VL.1996.545307.

[116] Else Swinnen and Carolien Toté. *NORMALIZED DIFFERENCE VEGETATION INDEX (NDVI)*. Tech. rep. Copernicus Global Land Operations, 2021.

[117] Espressif Systems. *ESP32 Bluetooth Architecture*. Tech. rep. Espressif Systems, 2019.

[118] Espressif Systems. *API Guides*. Tech. rep. Espressif Systems, 2021.

[119] K Takahashi, R Ooka, and S Ikeda. "Anomaly detection and missing data imputation in building energy data for automated data pre-processing." In: *Journal of Physics: Conference Series* 2069.1 (2021), p. 012144. DOI: 10.1088/1742-6596/2069/1/012144. URL: https://doi.org/10.1088/1742-6596/2069/1/012144.

[120] J.J. Thomas and K.A. Cook. "A visual analytics agenda." In: *IEEE Computer Graphics and Applications* 26.1 (2006), pp. 10–13. DOI: 10.1109/MCG.2006.5.

[121] Reet Kamal Tiwari and Akshar Tripathi. "AN UNSUPERVISED SOIL MOISTURE ESTIMATION MODELLING APPROACH USING C-BAND DUAL PolSAR DATA." In: (2020). DOI: 10.5194/egusphere-egu2020-12870. URL: https://doi.org/10.5194%2Fegusphere-egu2020-12870.

[122] Hong-Linh Truong, Tram Truong-Huu, and Tien-Dung Cao. "Making distributed edge machine learning for resource-constrained communities and environments smarter: contexts and challenges." In: *Journal of Reliable Intelligent Environments* (2022). DOI: 10.1007/s40860-022-00176-3. URL: https://doi.org/10.1007/s40860-022-00176-3.

[123] U.S Department of Labor. *Suspension Trauma/Orthostatic Intolerance*. Mar. 2011. URL: https://www.osha.gov/sites/default/files/publications/shib032404.pdf (visited on 06/23/2021).

[124] USHAIDI. *USHAIDI*. Website. URL: https://www.ushahidi.com/ (visited on 03/03/2021).

[125] United Nations. *17 Goals of Sustainable Development*. URL: https://sdgs.un.org/goals (visited on 03/13/2022).

[126] Radhakrishna Vangipuram, Rajesh Kumar Gunupudi, Veereswara Kumar Puligadda, and Janaki Vinjamuri. "A machine learning approach for imputation and anomaly detection in IoT environment." In: *Expert Systems* 37.5 (2020), e12556. DOI: https://doi.org/10.1111/exsy.12556. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/exsy.12556. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/exsy.12556.

[127] Andrés Villa-Henriksen, Gareth T.C. Edwards, Liisa A. Pesonen, Ole Green, and Claus Aage Grøn Sørensen. "Internet of Things in arable farming: Implementation, applications, challenges and potential." In: *Biosystems Engineering* 191 (2020), pp. 60 –84. ISSN: 1537-5110. DOI: https://doi.org/10.1016/j.biosystemseng.2019.12.013. URL: https://www.sciencedirect.com/science/article/pii/S1537511020300039.

[128] Antoni Vives. "Smart City Barcelona: The Catalan Quest to Improve Future Urban Living." In: *International Journal of Iberian Studies* 33.1 (Mar. 2018), pp. 103–104.

[129] Ne Wang, Ruiting Zhou, Lei Jiao, Renli Zhang, Bo Li, and Zongpeng Li. "Preemptive Scheduling for Distributed Machine Learning Jobs in Edge-Cloud Networks." In: *IEEE Journal on Selected Areas in Communications* 40.8 (2022), pp. 2411–2425. DOI: 10.1109/JSAC.2022.3180772.

[130] Carl Wellington, Aaron Courville, and Anthony (Tony) Stentz. "A Generative Model of Terrain for Autonomous Navigation in Vegetation." In: *The International Journal of Robotics Research* 25.12 (Dec. 2006), pp. 1287–1304. DOI: 10.1177/0278364906072769. URL: https://doi.org/10.1177%2F0278364906072769.

[131] Pete Wilson. "Smart communities guidebook." In: *Governor of California, CA, USA* (1997).

[132] Bernd W. Wirtz and Steven Birkmeyer. "Open Government: Origin, Development, and Conceptual Perspectives." In: *International Journal of Public Administration* 38.5 (Jan. 2015), pp. 381–396.

[133] Svante Wold, Kim Esbensen, and Paul Geladi. "Principal component analysis." In: *Chemometrics and Intelligent Laboratory Systems* 2.1 (1987). Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists, pp. 37–52. ISSN: 0169-7439. DOI: https://doi.org/10.1016/0169-7439(87)80084-9. URL: https://www.sciencedirect.com/science/article/pii/0169743987800849.

[134] F. Wu, T. Wu, and M. R. Yuce. "Design and Implementation of a Wearable Sensor Network System for IoT-Connected Safety and Health Applications." In: *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. 2019, pp. 87–90. DOI: 10.1109/WF-IoT.2019.8767280.

[135] Dvoralai Wulfsohn, Felipe Aravena Zamora, Camilla Potin Téllez, Inés Zamora Lagos, and Marta García-Fiñana. "Multilevel systematic sampling to estimate total fruit number for yield forecasts." In: *Precision Agriculture* 13.2 (2012), pp. 256–275.

[136] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauly, Michael J. Franklin, Scott Shenker, and Ion Stoica. "Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing." In: *USENIX Symp. on Networked Systems Design and Implementation*. 2012, pp. 15–28.

[137] Matei Zaharia et al. "Apache Spark: a unified engine for big data processing." In: *Communications of the ACM* 59.11 (2016), pp. 56–65.

[138] Zhaoyu Zhai, José Fernán Martínez, Victoria Beltran, and Néstor Lucas Martínez. "Decision support systems for agriculture 4.0: Survey and challenges." In: *Computers and Electronics in Agriculture* 170 (2020).

[139] Zhi Zhou, Xu Chen, En Li, Liekang Zeng, Ke Luo, and Junshan Zhang. "Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing." In: *Proceedings of the IEEE* 107.8 (2019), pp. 1738–1762. DOI: 10.1109/JPROC.2019.2918951.