People’s Democratic Republic of Algeria
Ministry of Higher Education and Research
Saad Dahleb University of Blida –1
University of Medea

A Master's Thesis

Submitted to the computer science department
In fulfillment of the requirements
For an Information systems and networks' degree

Title

Cooperative and low-speed network-tolerant Geographic Information System (GIS) to support disaster recovery

By
Bentalbi Katia

Directed by
Dr. Allal Tiberkak

2023-2024

Abstract

The necessity for efficient disaster management systems has become more apparent as a result to the escalating occurrence and gravity of both natural and man-made disasters across the globe. Traditional systems often grapple with communication bottlenecks and limited accessibility, impeding prompt and coordinated responses during emergencies. This thesis proposes an innovative strategy to address these challenges by developing an advanced disaster management system that integrates Geographic Information System technology with Edge Computing architecture and Virtual Reality capabilities.

By harnessing the synergistic potential of Geographic Information System, Edge Computing, and Virtual Reality technologies, this transformative approach aims to enhance communication, spatial visualization capacities, and decision-making processes. Through seamless integration, the system seeks to bolster disaster response efforts on larger scales. Ultimately contributing to the resilience and safety of communities countrywide.

Résumé

La nécessité de mettre en place des systèmes efficaces de gestion des catastrophes est devenue plus évidente en raison de l'augmentation de la fréquence et de la gravité des catastrophes naturelles et d'origine humaine dans le monde entier. Les systèmes traditionnels sont souvent confrontés à des goulets d'étranglement en matière de communication et à une accessibilité limitée, ce qui empêche d'apporter des réponses rapides et coordonnées en cas d'urgence. Cette thèse propose une stratégie innovante pour relever ces défis en développant un système avancé de gestion des catastrophes qui intègre la technologie du système d'information géographique avec l'architecture Edge Computing et les capacités de la réalité virtuelle.

En exploitant le potentiel synergique des technologies Système d’information géographique, Informatique en périphérie de réseau et Réalité Virtuelle, cette approche transformatrice vise à améliorer la communication, les capacités de visualisation spatiale et les processus de prise de décision. Grâce à une intégration transparente, le système proposé cherche à soutenir les efforts de réponse aux catastrophes à plus grande échelle. En fin de compte, il contribuera à la résilience et à la sécurité des communautés dans tout le pays.

الملخص

لقد أصبحت الحاجة إلى نظم فعالة لإدارة الكوارث أكثر وضوحاً نتيجة لتصاعد نسبة حدوث وخطورة الكوارث الطبيعية والكوارث التي من صنع الإنسان في جميع أنحاء العالم. وغالباً ما تعاني الأنظمة التقليدية من اختناقات في الاتصالات ومحدودية إمكانية التوفر من حيث المعلومات الضرورية، مما يعيق الاستجابات السريعة والمنسقة أثناء حالات الطوارئ. تقترح هذه الأطروحة استراتيجية مبتكرة لمواجهة هذه التحديات من خلال تطوير نظام متقدم لإدارة الكوارث يدمج تكنولوجيا نظام المعلومات الجغرافية مع بنية حوسبة الحافة وقدرات الواقع الافتراضي.

ومن خلال تسخير الإمكانات المتآزرة لتقنيات المذكورة، يهدف هذا النهج التحويلي إلى تعزيز التواصل وقدرات التصوير المكاني وعمليات صنع القرار. ويسعى النظام من خلال التكامل السلس بين المبادئ، إلى تعزيز جهود الاستجابة للكوارث على نطاق أوسع. مما يسهم في نهاية المطاف في تعزيز مرونة وسلامة المجتمعات في جميع أنحاء البلاد.

- **Keywords :** Disaster recovery, Cooperative platform, Geographic Information System, Cloud/Edge/ Fog/ computing, Virtual Reality.

Acknowledgements

I would like to express my sincere gratitude to my parents for their encouragement throughout my academic path. Their dedication to offering the necessary resources has been incredibly instrumental in the completion of this thesis. I am deeply grateful for your unwavering support and constant presence in my life.

I also extend my thanks to my advisor, Dr. Allal Tiberkak for their invaluable guidance, feedback, adding more depth into my researches, and allowing me to take more requirements into consideration that allowed me to reach higher standards.

Finally, I would like to acknowledge my friends Testas Dounia and Fertoul Meroua for their continuous encouragement and support.

Thank you all for your support and contributions.

Bentalbi Katia

Table of content

Chapter One: Generalities

Chapter Two: Related Works

Chapter three: Methodology

Chapter four: Results

List of Figures

List of Tables

List of abbreviations

IS : Information System

GIS : Geographical Information System

EC : Edge Computing

MEC : Multi-access Edge Computing

VR : Virtual Reality

MR : Merged Reality

AR : Augmented Reality

AI : Artificial Intelligence

ML : Machine Learning

BIM : Building Information System

DSM : Digital Surface Model

VPF : Virtual Permitting Framework

F-Fictional : Fake Fictional

O-Fictional : nOn genuine Fictional

ABC : Artificial Bee Colony algorithm

FC : Fog Computing

FCN : Fog Computing Nodes

DMS : Disaster Management System

DRM : Disaster Risk Management

WLAN : Wireless Local Area Network

RAN : Radio Area Network

WSAN : Wireless Sensor and Actuator Network

IOT : Internet Of Things

IOV : Internet Of Vehicle

DTM : Digital Terrain Modeling

RS : Remote Sensing

P-Real : Physical Real

G-Real : Genuine Real

3D : three-Dimensional

Introduction

Our society is always in the pursuit of progress, exploiting new technologies and concepts to make life more convenient. But it can be slowed down by natural disasters, which can cause numerous negative impacts [1], such as incapability of transmitting accurate information on the disaster, economic setbacks, or infrastructure destruction and humanitarian crises that can be defined by the difficulty of distinguishing and providing health care or in another term being unable to determine the safe zones from the dangerous.

Going from this point we can express the progress by the desire to evolve as a society without being impoverished by disasters that can threaten this process. Besides that, traditional disaster management systems often face limitations in data processing speed, real-time information dissemination, and visualization capabilities. This necessitates exploring advanced technologies to improve efficiency and effectiveness. That’s why our studies will treat the development of a disaster management system with geographical information system (GIS), an information system that process geospatial data and generate the results in the appropriate formats. The GIS will be based on virtual reality (VR) to provide a more realistic and interactive representation of the map. This integration will respect the architecture of multi-access edge computing (MEC) that will get data processed quickly which later will allow us to take decisions in a short amount of time when we’ll have to deal with urgent situations like natural disasters.

- **State of art:** Current knowledge and Objectives.

The increasing complexity and frequency of disasters pose a significant challenge to existing management systems. This section provides a comprehensive overview of the current state of these technologies. However, many of them face limitations that hinder effective response. These limitations often involve a reliance on centralized data processing, which can become overloaded or inaccessible during critical moments.

Additionally, static data representation, where information isn't updated in real-time, restricts situational awareness and hampers timely decision-making. To address these shortcomings and equip disaster response teams with the necessary tools, we require more sophisticated technologies. These technologies should be real-time, dynamic, and interactive. Real-time data allows for constant updates on the evolving situation, providing a more accurate picture of the disaster's impact. Dynamic systems can adapt to changing circumstances and offer relevant insights to guide response efforts. Finally, interactive systems facilitate collaboration and communication between various stakeholders, ensuring coordinated and efficient disaster management.

Edge Computing is becoming significantly important in the realm of disaster management. By handling data in close proximity to its origin, it mitigates delays and improves the dependability of data transmission This decentralized approach ensures that critical data remains accessible and actionable even when central communication networks are disrupted. Preceding to Geographic Information Systems that are indispensable in the creation of detailed maps and models essential for disaster preparedness, response, and recovery. It enables the integration and analysis of spatial data, providing critical insights into the geographic and environmental aspects of disasters.

Wrapping up with Virtual Reality that offers transformative potential for disaster management by providing an immersive platform for simulation and visualization. For first responders and decision-makers, VR can create virtual replicas of disaster-affected areas, allowing for detailed exploration and scenario planning without the risks associated with physical presence. This capability enhances situational awareness and facilitates the development of more effective intervention strategies.

- **Proposed Solution and Results:** Methodology and Findings.

Addressing the solution from a front-end, and back-end perspective, the front will be built on GIS and VR technologies, allowing users to visualize disaster scenarios in a three-dimensional space, offering a more intuitive and comprehensive understanding of the affected areas. This interface will allow users to interact with and create, sharing data in real-time, enhancing their ability to respond to disaster scenarios dynamically. In addition, the ability to overlay various data layers, such as infrastructures, and hazard zones, allows for nuanced analysis and informed decision-making but also the possibility to add and delete data that is purely processed during the disaster without letting it affect the layers that are part of the base map.

As for the backend, powered by EC, will ensure that data processing and analytics are performed efficiently and reliably, providing a resilient infrastructure that can withstand the uncertainties of disaster zones. Moreover, it will contribute into enhancing the representation of the front end, by splitting data and having the privilege to load high resolution imagery in each edge, web map created.

Ending by the main requirement that should be taken into consideration is ensuring communication, which means the users will be able to share spatial data within their group of the same edge, but also transmit it to users of different group of another edge to get notified and easily control any disaster scenario in a short time.

- **Structure of the Thesis:** Overview of chapters and Sections.

Chapter one outlines the primary objectives of this research, which center around enhancing communication efficiency and leveraging cutting-edge technologies to bolster disaster response capabilities. Additionally, Chapter One introduces the proposed solution and methodology, offering a glimpse into the anticipated outcomes and contributions of this study.

Chapter two presents essential concepts that form the basis of the proposed disaster management system It commences by explicating the core principles of disaster management systems, delineating their significance in disaster preparedness, response, and recovery initiatives. Subsequently, delving into Edge Computing and its various implementations such as Multi-access EC, Fog Computing, and Cloudlet, along with their potential impacts on disaster management. Moreover, it scrutinizes Virtual Reality technology, including Augmented Reality and Mixed Reality, and their transformative capacity in improving the visualization and understanding of disaster scenarios. Finally, the chapter investigates the fundamental elements of Geographic Information System, outlining its user layer, functional layer, information layer, and communication layer, while emphasizing its crucial role in spatial data analysis and visualization.

Chapter Three embarks on a comprehensive review of related works in the field, aiming to contextualize the proposed solution within the broader research perspective. It demarcates the fusion of GIS, VR, and EC technologies, exploring existing models and frameworks for integrating these constructs to enhance disaster management capabilities. Moreover, the chapter explores recent three-dimensional GIS systems and the emergence of cutting-edge GIS disaster management systems which utilizes sensor data, artificial intelligence paradigms, and remote sensing methods to enhance decision-making. Furthermore, it presents a summary of recent scholarly efforts centering on urban disaster management systems, highlighting the incorporation of Building Information Modeling and GIS for thorough disaster risk evaluation and management.

Chapter Four delineates the methodology employed in developing the proposed disaster management system, contextualizing the research within the broader disciplinary framework and outlines the communication strategies adopted to facilitate collaboration and information dissemination among users. Besides, system requirements are elaborated, detailing the development process of the GIS and the creation of the web application interface. Additionally, it offers comprehensive explications regarding the system's structure, including use case diagrams and sequence diagrams, to clarify its operational capabilities.

Chapter One: Generalities

I. Introduction

Generalities serve as a building block, providing context, terminology, and conceptual frameworks that underpin subsequent discussions. Establishing a foundational understanding of certain key topics is essential. Beginning with disaster management, to get familiarized with the different phases that constitute a traditional emergency response system.

Following by Edge Computing, a paradigm that brings data processing closer to the source, enabling faster response time and reduced reliance on centralized Cloud Computing. Continuing the discussion by Virtual Reality (VR), capable of creating immersive simulated environments and simulate diverse disaster scenarios, providing invaluable training and preparedness opportunities for emergency responders.

Lastly, delving into Geographical Information Systems, robust tools designed for capturing, storing, analyzing, and visualizing spatial data. GIS technology plays a crucial role in disaster management by providing valuable insights into geographic resource, allocation, and risk assessment.

By synthesizing insights from these diverse technological realms, the chapter aims to lay the groundwork for a solution that promises to revolutionize disaster management field. Through the strategic integration of EC, VR, and GIS, it aspires to cultivate a more resilient, adaptive, and efficient disaster management ecosystem, poised to safeguard lives and minimize property damage in the face of adversity.

II. Disaster management system [1]

Natural disasters, such as earthquakes, wildfires, and viruses, present a significant obstacle to public safety and government organizations responsible for managing disasters. Instances of failure in responding to disasters can easily occur due to mishandling of the situation or the use of an ineffective disaster management strategy that’s why a disaster management system is a necessity to every organism in to avoid all negative impacts.

Disaster management is comprised of several phases, including risk assessment, mitigation, readiness, response, and recovery. Each phase is interconnected and comprises a set of activities that collectively aim to reduce the negative impacts of loss that can result from disasters. (Figure 1)

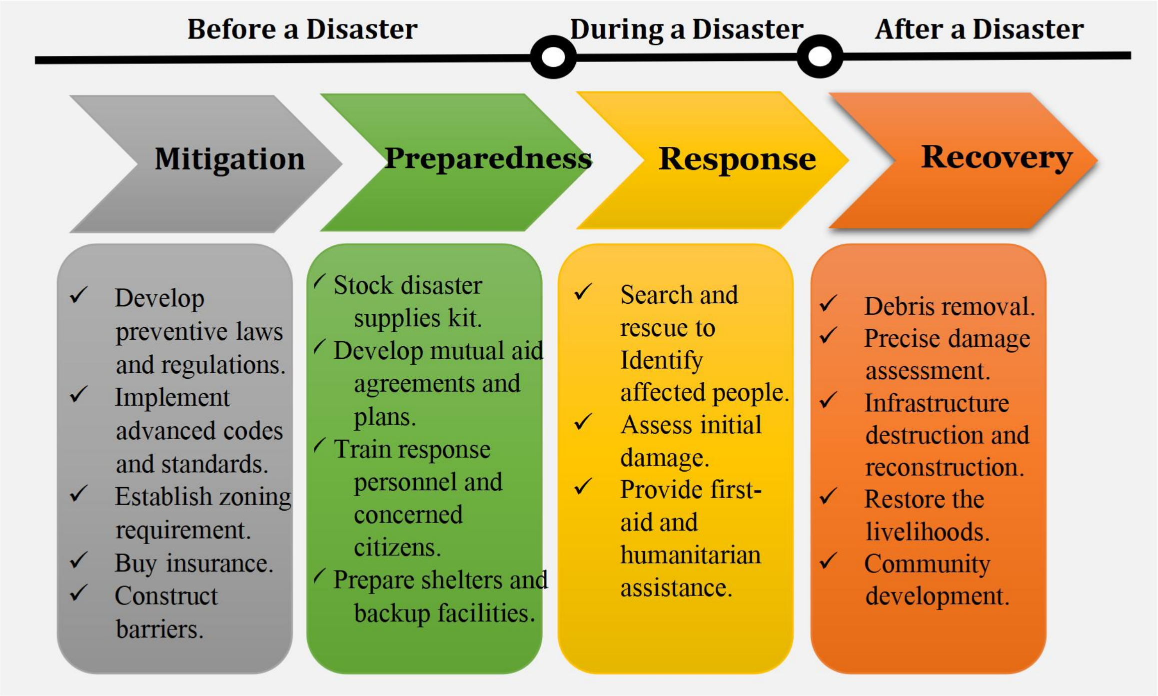


Figure 1: *Disaster management phases [7] [17]*

However, there are ongoing challenges in coordinating resources, managing data, and addressing environmental factors that contribute to the occurrence of disasters. By covering each disaster to define its potential causes, it will allow us to highlight possible methodologies and findings to outline future research directions while developing our disaster management system.

III. Edge Computing (EC):

With the emergence [22] of 5G systems and IoT, the number of connected devices continued to increase exponentially, traditional cloud paradigms struggled to efficiently allocate resources for tasks, leading to concerns about security, storage, computational performance, reliability, and big data storage. Additionally, the need to fulfill requirements of mobility, responsiveness, and scalability that was imposed by IoT applications had to be taken into consideration.

That is why edge computing emerged as a solution [10], its approach consists of placing computing resources at the edge in proximity to mobile devices. Edge computing came to give priority and quality to computation, allows storing critical more frequent used data at the edge to let this last get processed faster, which goes against the trend of consolidation seen in traditional cloud computing.

- Implementations [22]

Within the realm of edge computing, several implementations have been developed to cater diverse use cases and operational requirements. Multi-access Edge Computing, Fog computing and Cloudlet represent are among the most significant, each offering unique benefits and addressing specific challenges. making EC an integral part of modern data processing and service delivery frameworks.

- **Multi-access Edge computing (MEC):** Computing resources are extended to the periphery of the Radio Access Network (RAN) (Figure 2). MEC nodes are typically deployed alongside macro base stations or radio network controllers and support various host instances for storage and computation. These hosts utilize virtualized interfaces like virtual machines or containers and are managed by a hypervisor, which oversees resource allocation and application management based on network topology and available resources.

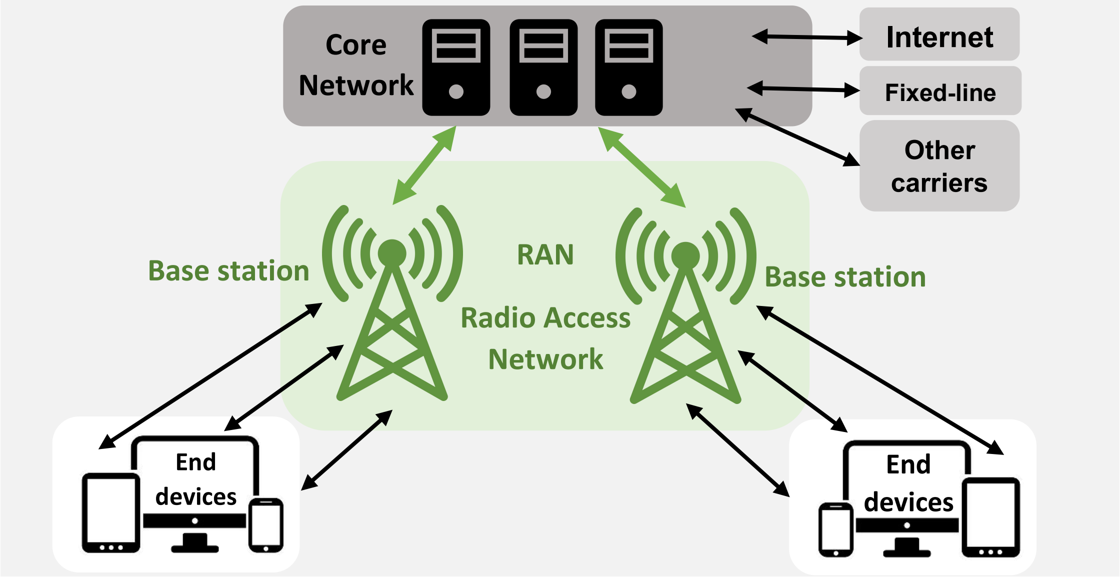


Figure 2: Basic Architecture of RAN [24]

- **Fog Computing:** Deployed between edge devices and Clouds to create a decentralized computing infrastructure, Fog Computing Nodes (FCNs) can incorporate various device types such as switches, routers, IoT gateways, access points, providing services to the diverse needs. They also support devices across different protocol layers and are compatible with non-IP-based access technologies. FCNs operate transparently to Edge equipment through a common Fog abstraction layer, offering functionalities like resource assignment, security, monitoring, and equipment management. The service orchestration layer utilizes these functions to receive requests from end users and allocate resources accordingly, just as Figure 3 demonstrate below:

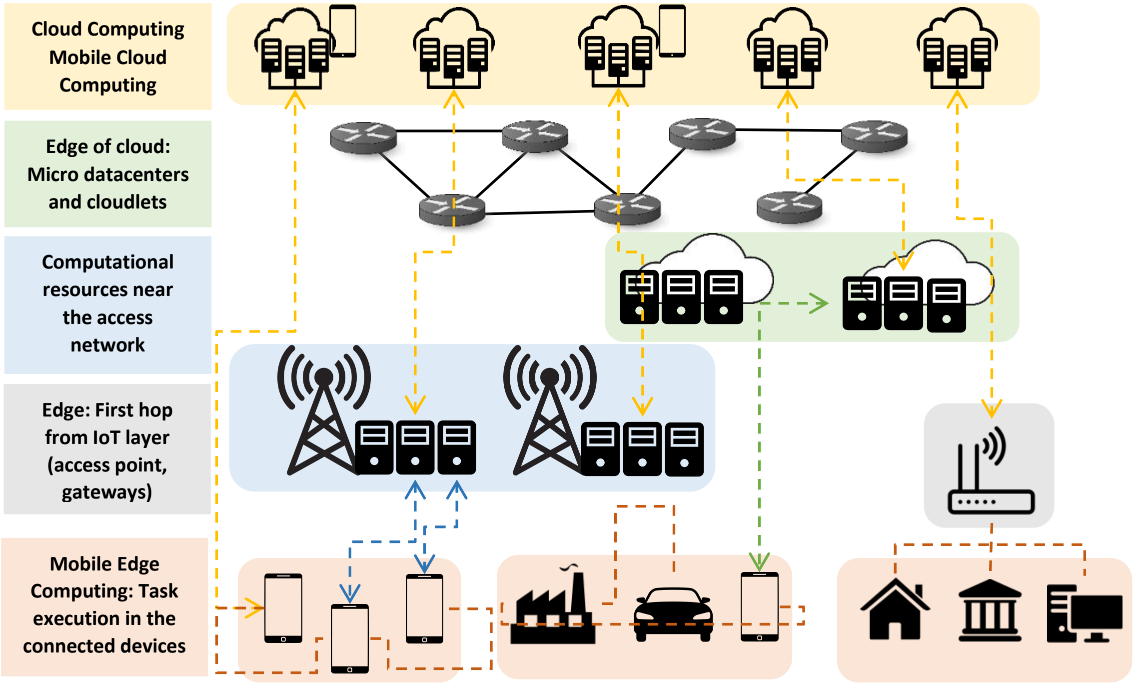


Figure 3: How Cloud, Fog and Edge computing function in processing and allocating data resources according to end users requests [25]

The co-deployment of fog and Internet of things (IoT) enables real-time monitoring and analysis of data, with fog nodes serving as processing and storage points for IoT-generated data. Fog computing benefits IoT deployments by enhancing cognition, efficiency, flexibility, thereby supporting diverse IoT scenarios like Internet of Vehicles (IoV), Wireless Sensor and Actuator Networks (WSAN), and Smart Grids.

- **Cloudlet:** A Cloudlet is a cluster of multicore processors that offers services with high bandwidth within a one-hop access range, such as end users and mobile devices via WLAN networks, resulting in low latency. The architecture of a Cloudlet consists of three layers: the Cloudlet layer, the node layer, and the component layer. Starting from the bottom, the component layer provides interfaces to the upper layers. The Cloudlet layer comprises a cluster of co-deployed nodes, each equipped with one or more execution environments. While node layer is comprised of node agents that manage the nodes, and Cloudlet agent who supervises the Cloudlet layer.

The edge [9], which is located one hop away from end devices, supports various applications such as augmented reality, public safety, autonomous driving, smart manufacturing and healthcare.

In terms of functions, edge computing involves a two-way computing stream:

- Upstream (from the devices to the cloud)
- Downstream (from the cloud to the devices)

Taking into consideration that the design of the edge should be crucial to meet requirements, and end devices do not only consume data but also produce it, allowing for tasks such as computing offloading, data storage, cashing, processing and service delivery from the cloud to the user.

Another important concept should be brought that is three-tier edge computing (Figure 4), the model consists of IoT, edge and cloud. IoT devices such as drones and sensors, connect to the edge using various communication protocols. The edge, which includes connected vehicles, cellular tower, gateways and servers, relies on cloud computing capabilities for complex tasks, with specific communication protocols between IoT-edge and edge-cloud.

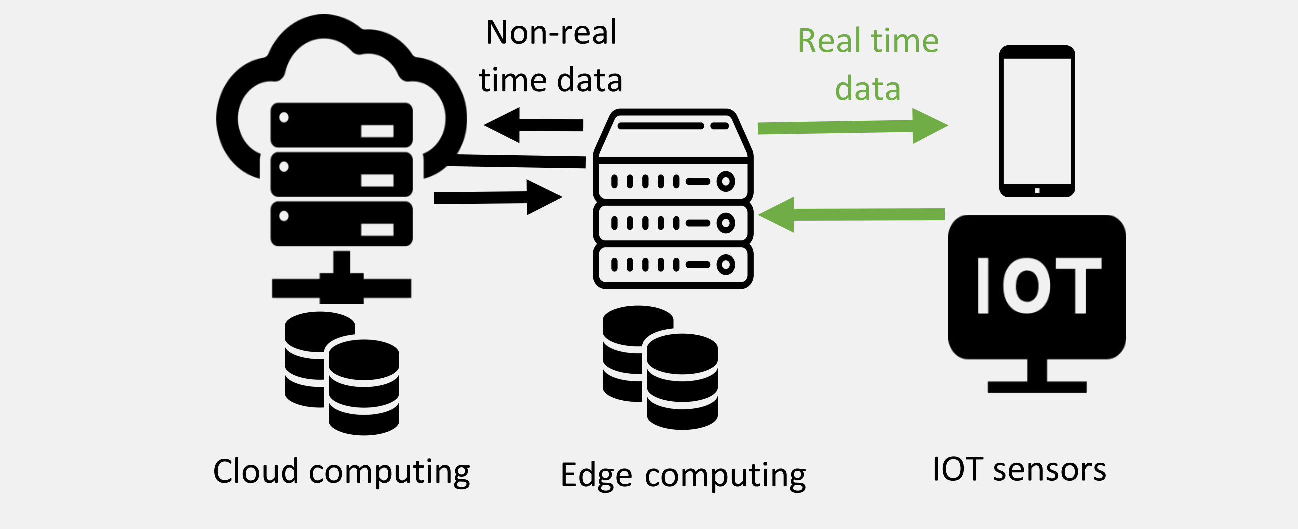


Figure 4: *Three-tier edge computing and how data is processed.*

To treat the main debate if edge computing is better than cloud computing and the answer is that they complement each other. Edge computing handles temporary data at the network edge, reducing network bandwidth pressure and data center power consumption. This decentralized processing also minimizes system latency and improves service response capability. Notably, edge computing stores the private data on edge devices, mitigating the risk of network data leakage and ensuring security and privacy.

Despite the highlighted potential of Edge Computing, numerous challenges remain across fundamental technologies, application scenarios, and business models. Weisong Shi et all (2019) [9] dedicated a section to the state of the art in edge computing, highlighting significant progress observed in various areas over the past five years. This focus aims to advance understanding and foster innovation in the field of edge computing.

Despite bringing computation closer to delay-sensitive services, challenges persist due to the rising pace of data generation and the dynamic nature of edge nodes, which can be mobile and subject to connectivity failures and bandwidth fluctuations.

The key research challenges include orchestrating edge services, addressing resource heterogeneity, managing service lifecycle, optimizing task offloading, and scheduling data management and processing tasks efficiently.

IV. Virtual reality (VR):

Exploring "virtual reality" [23] reveals a paradox between "virtual" and "real." Traditionally, "virtual" implies potential existence, but in the digital age, it merges with fictionality, causing conceptual confusion. Scholars like Chalmers introduces "virtual realism," suggesting virtual objects possess distinct ontological status. Debates persist regarding which objects can be successfully virtualized. Critics, however, advocate for "virtual fictionalism," a perspective that questions the genuine reality of virtual objects, arguing that despite their functional presence, they lack the same ontological weight as physical objects.

To avoid misunderstandings, clarifying the meanings of "virtual," "real," and "fictional" is crucial. "Virtual" objects are digitized representations preserving functionality, reframing discussions to "virtual environments" clarifies the nature of virtual experiences. "Real" is distinguished between "p-real" for physical existence and "g-real" for authenticity, while "fictional" is categorized into "o-fictional" for fictional entities and "f-fictional" for fake or non-genuine objects. These distinctions expose flaws in naive virtual unrealism and virtual fictionalism.

Critiquing the use of Waltonian fictions[1], virtual environments resemble playing with toys rather than make-believe. Toys offload mental processes by representing simplified versions of genuine objects. Virtual environments provide immersive experiences without relying on make-believe.

The stark division between virtual and physical objects is challenged by the concept of "virtual physicalism." Virtual objects are classified as either genuine (g-real) or fake (fictional). Concerning fiction, it is argued virtual objects are not inherently fictional (o-fictional), even when part of narratives, ultimately remaining digital toys.

Virtual objects are considered real, existing within the physical realm of computer states, and their ontological status depends on functional similarity to non-virtual objects. Integrating virtual environments into reality could expand the perception of virtual objects as genuine entities. It is suggested that integrating virtual environments more deeply into reality, making them persistent and causally connected. The successful virtualization of artifacts is discussed, along with questions about virtualizing natural kinds and individuals, intertwining with broader philosophical discussions [23].

To add more context and get better understanding on Virtual Reality (VR) we're going to introduce some more key concepts:

- **Augmented Reality (AR):** Unlike (VR), Augmented reality doesn't replace the real world, it overlays digital information, such as images, videos or 3D models where users can see both physical environment and digital content through smartphones, tablets, or dedicated AR eyewear to facilitate the integration of virtual components in their surroundings in a synchronous way.
- **Mixed Reality (MR):** Combines both VR and AR, it can be defined by the convergence of digital objects and let them interact with the physical environment. The main difference between AR and MR though they seem both alike, but MR incorporate spatial awareness, creating a more interactive experience for the users.

VR, AR and MR may have distinct experiences, but they all have the same goal into trying to alternate the perception through technology to create an immersive and interactive

---

[1] Concept in philosophy of fiction developed by Kendall Walton, Waltonian fictions are works of fiction that create possible worlds within narrative, even though we know they are not real.

experience. Which the progress of one of these technologies invariably drives the progress of the others.

V. Geographic information system (GIS):

Geographic information systems (GIS) [2] is a type of information system (IS) that involves various data operations such as collection, storage, integration, analysis and graphical interpretation of spatiotemporal, along with associated attribute information. The data sources for GIS include cartographic materials, remote sensing data, geodetic measurements, state statistical services, stationary measuring observation posts, crowd sourcing data and geodata from open sources.

A GIS general architecture [13] is composed of multiple layers; each layer interacts with the others, providing a certain service for the GIS to be functional, just as shown in (Figure 5):

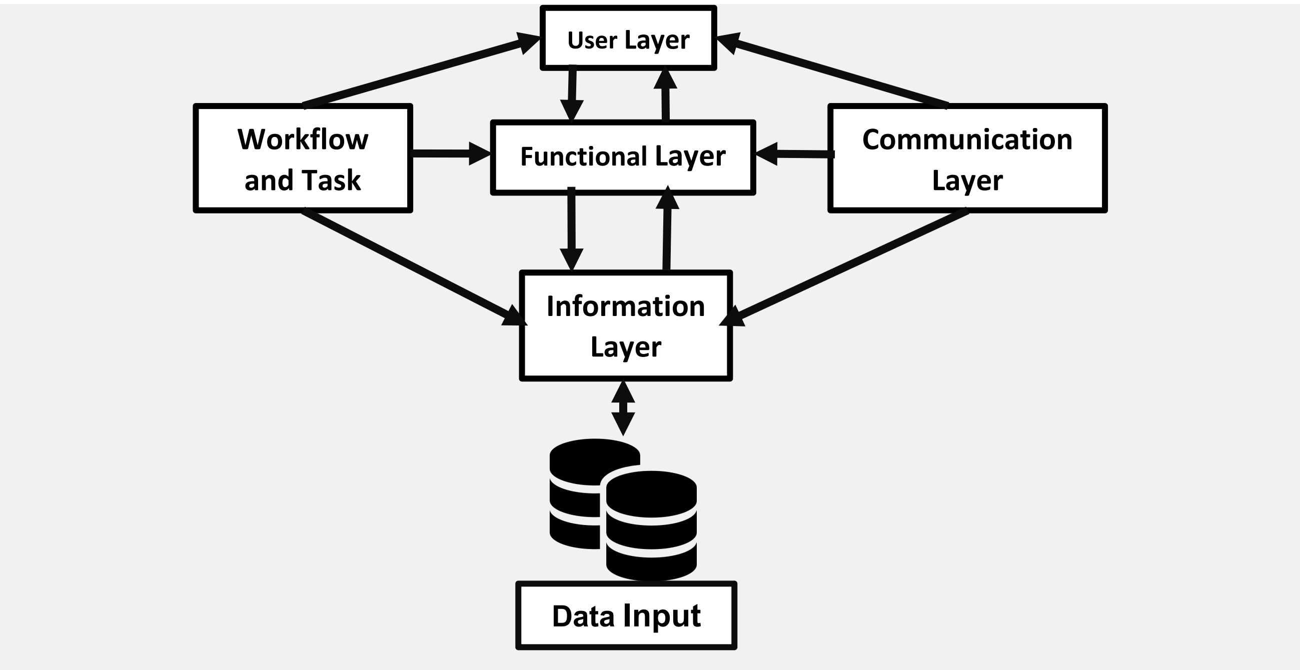


Figure 5: Architecture of a geographical information system (GIS) [15]

- **User layer:** Enables website users to interact with a GIS through a browser, including web services and databases.
- **Functional layer:** Contains software components facilitating interaction between the user interface and the geospatial database.
- **Information layer:** Stores geospatial data used for map creation and spatial analysis.
- **Communication layer:** Processes and analyzes geospatial data, visualizes results, and facilitates data export.

The functioning scheme of GIS can be unified by

- Compromising input source data, visualization, spatial analysis, and modeling.
- Visualization and editing results, and output of results to users in supported formats.

It is also structured with a core, subsystem for importing/exporting data, tools for functionality expansion, and graphical user interface.

A **web-based GIS** is implemented as client-server application where the client (browser) manages the user's interface and requests, while the server handles data storage, protection, access and generate the response just as shown in (Figure 6). Clients are classified into two categories, thin clients are lightweight applications or devices that rely heavily on the server for processing and data storage, primarily using standard web browsers to access applications hosted on the server. In contrast, thick clients are standalone applications installed locally on the user's device, performing a significant portion of processing and data storage locally.

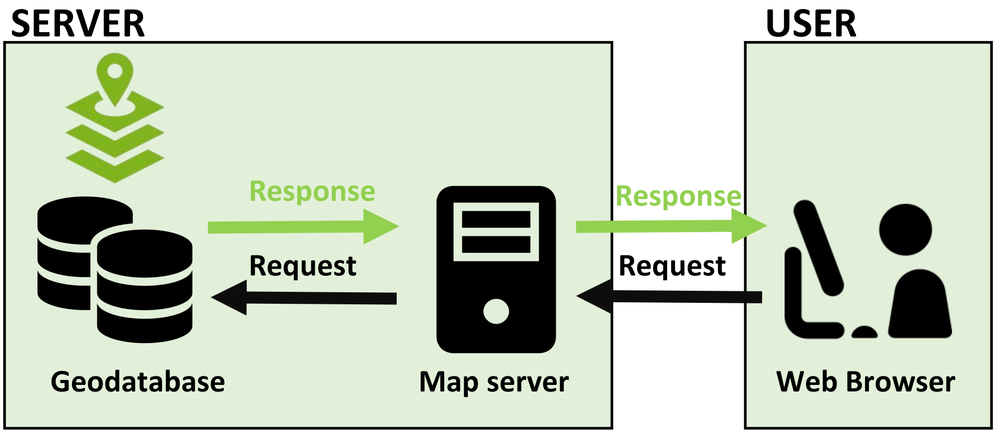


Figure 6: Architecture of a web-based GIS [14]

VI. Conclusion

In conclusion, this chapter offers an exhaustive overview of the current state of integration among Edge Computing, Geographic Information Systems, Virtual Reality, and Disaster Management Systems. Through detailed sections explaining the context and terminology associated with each technology, a deeper comprehension of their individual functionalities and potential synergies has been attained.

By investigating the convergence of these technologies, the groundwork has been laid for exploring innovative solutions aimed at enhancing disaster preparedness, response, and recovery endeavors. VR-GIS integration enables the visualization of diverse scenarios, facilitating decision-making and error prevention, while EC facilitates real-time data processing at disaster sites. Nonetheless, several gaps persist, including limitations in implementation, user interaction, scalability, and adaptability.

Consequently, further research is imperative in the subsequent chapter to develop a responsive disaster management system. This entails an examination of existing and recent systems, delineation of their methodologies, and identification of pertinent gaps for potential mitigation and resolution.

Chapter three: Related works

I. Introduction

This chapter outlines various studies that employ the technologies previously defined. After establishing the foundational concepts and notions, the next step is to explore recent findings. Each section presents and analyzes individual works, examining their strengths and limitations to provide a clearer view and the direction the solution will take. This includes identifying the methodologies employed, research techniques used, and categorizing the work according to its relevance to the study problem.

The initial section is devoted to the integration [2] of GIS, VR, and EC. It discusses theoretical studies and the potential of combining them. It also introduces the architectural framework proposed by edge computing for three-dimensional GIS, which enhances the realism of the representation. On the other hand, the next section will propose a system based on large-scale 3D GIS [3], moving beyond the theorical discussion and focusing on execution and results.

Subsequently, the chapter will discuss recent GIS disaster management systems [4, 5, 7] where remote sensing and artificial intelligence are introduced and integrated. The objective will be to gain a deeper understanding of the requirements for effective disaster management systems. It concludes with an overview of all the selected works, highlighting key findings and transitioning to the next chapter, where the methodology will be developed.

II. Fusion GIS, VR and EC

Fei Wang et al (2021) [2] highlighted the challenges that arise when constructing and managing digital cities within the broader framework of digital earth. This fundamental framework relies on spatial information systems, specifically geographic information systems (GIS). However, as geographic information science progresses, working within a two-dimensional representation starts bringing limitations, resulting in growing demand to use three-dimensional (3D) representation.

The paper put emphasis on the importance of 3D GIS in providing a more realistic depiction of spatial phenomena. This enables spatial analysis and operation of complex spatial objects in various sectors such as land management, urban planning, and infrastructure.

The fusion between virtual reality and geographical information systems is recognized as an emerging trend that seeks to overcome the limitations in traditional spatial representations by facilitating the creation of a more realistic representation of geographical data, in which will allow offering the user a more immersive experience.

Nevertheless, to tackle the obstacles and difficulties that might arise due to managing more significant data while developing the representation, this paper introduces the notion of edge computing (EC) as a potential solution. EC involves processing data at the edge, near the application scenario which it reduces latency and delay. Its integration to the previous technologies aims to strike a balance between enhancing the user experience of VR and managing costs efficiently.

Going through the different related studies that have been mentioned, the first limitation that have been brought are traditional GIS applications and how far from reality their three-dimensional representation can be. As a solution, many approaches have been explored, like automatically constructing intricate and large-scale 3D road networks based on GIS [3] by using satellite imagery, elevation date and 2D road median data. Other hybrid systems have been designed for multiple purposes, such as extensive terrains, displaying both in 2D and 3D, extending 2D data analysis functions of traditional geographical information systems to 3D environments with 3D geographical data. It is cited that ever since virtual reality got integrated to GIS, the progress has been achieved in implementing multi-view approaches that connects both technologies, but when it comes to manage heavy or complex operations such as processing data while assuring availability and efficiency, here comes the role of edge computing.

- **Modeling:**

In this section, the author treated different aspects on "how to develop a solution" starting by **system modeling** that presents different components of the system and how they interact with each other:

- ✓ A GIS consisted of hardware, software, system management operators and geographic or spatial data that represents the content.
- ✓ A 3D GIS that relies on virtual reality (VR) sensing devices and edge computing (EC) with edge servers located on a proximity to users.

All summarized within the following illustration (Figure 7) showcasing one central server that stores all geographic information, multiple edge servers that process specific geographic data. End users with VR devices to request to visualize geographic information. Intermediate nodes to facilitate the transmission of requests and geographic information.

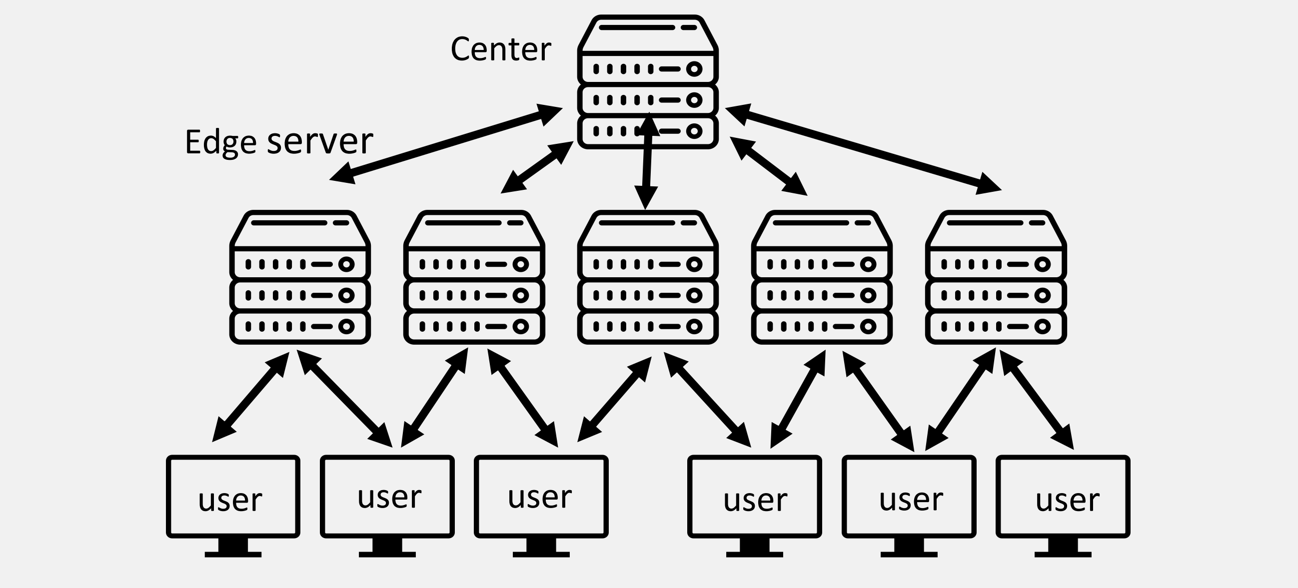


Figure 7: Geographical Information system (GIS) framework [2]

Then comes **problem modeling** where they focused on the optimization of GIS service quality and cost reduction. The main goals of optimization were centered around minimizing performance metrics such as latency, waiting time for requests, and response delay. So, in order get a better understanding on our problem, calculations have been made like the total delay of system that is calculated as the sum of delays experienced by individual users.

In term of cost, the analysis considers both financial and energy consumption factors. Financial costs include expenses associated with deployment of VR devices and edge servers. When energy costs, on the other hand, encompass the inherent hardware consumption computational energy. So, the total cost of the system can be calculated by summing up the costs incurred by individual users.

The optimization of the problem can be formulated by an objective function:

$$Min\ P = \alpha\ Delay + \beta\ Cost$$

Where α and β are the weight factors of each performance metric, respectively; they reflect the comprehensive consideration of performance in the optimization process.

More experiments were proposed to establish and prove the efficacy that the integration of GIS, VR and EC can provide, these experiments involved a comparison between three existing systems a 3D GIS platform construction of expressway based on 3D data model [11], GIS-Based 3D modeling and visualization in Luohe Basin [12] and of course a system based on GIS, virtual reality and edge computing. Different aspects were the subject of the study like comparing results on delay and cost, but before presenting any result we must acknowledge how experimental data is extracted to develop a system.

The experimental data must have three components:

- ✓ a digital map that refers to geographical data such as roads, buildings, bodies of water, plants, and other ground data.
- ✓ a scene entity that refers to characteristics and structure within the three-dimensional model and representation.
- ✓ a texture data that refers to real imagery from field photography.

All those three components are collected and processed to illuminate inaccuracies, unnecessary information and redundancies, all while maintaining the original data's accuracy. By using the previous performance equations that were deducted while modeling the problem itself they managed to get the following results:

- **Delay:** calculated by summing the set of time metrics for how many numbers of operations between two nodes, the comparative experimental result of VR EC GIS, 3D GIS data model and GIS 3D model and visualization is that the system using VR, EC, GIS had the lowest delay followed by 3D GIS data model then GIS 3D model and visualization.

The delay results put an emphasis on Edge computing and how it can allow to get quick response, quick data processing avoiding sending the request to distant servers and let them be handled by edge servers, thereby reducing transmission delay.

- **Cost:** The cost was calculated just as mentioned before summing all the costs of every individual user and the comparative experimental outcomes of the three systems were the same, one more time the VR EC GIS integrated system had the lowest cost followed by 3D GIS data model then model and visualization.

This signifies that 3D GIS construction with VR and EC surpasses the other two algorithms in term of cost.

The study puts forward a GIS based on virtual reality and edge computing, but also recommends an enhanced artificial bee colony (ABC) algorithm to acquire the optimal deployment plan of edge servers in the system considering delay and cost as optimization goals.

III. Three-dimensional (3D) GIS:

Hua Wang et al (2021) [3] highlighted on the importance of three-dimensional (3D) virtual road network for immersive traffic controls. The main challenge lies in creating a realistic large-scale 3D road network that includes a lifelike scene, smooth road surfaces, and a traffic semantic structure. They also mentioned existing methods that consisted of either merge 3D models into a unified view or use satellite imagery for texture but encounter difficulties in data denoising, texture reconstruction and generating different intersections. To overcome those challenges, a solution based on open-source geographic information system (GIS) data, introduces a semantic description of road networks, generates 2D road shapes automatically and use piecewise B-spline curves to fit elevation data, resulting in a detailed 3D virtual road network.

To have concrete bases on how to develop the solution, the author split the problematic into two main parts that are how model a 3D scene and a traffic network. Many studies were mentioned to cover each part starting off by 3D scene modeling. Hua wang cited the work of both Garciadorado et al (2017) and parish et al (2001) who automated the creation of roads in 3D scenes, by incorporating lane attributes, intersections, and road shapes or using a GIS software. And while these scenes showcase intricate objects like buildings and trees, it was difficult to achieve consistent textures with actual environment. Dias et al (2006) tried to use laser scanners for presided 3D modeling but encountered the same challenges in denoising the data and reconstructing textures.

However, there were more effective methods in generating 3D road networks based on GIS data, such as satellite imagery and elevation models. A. Agrawal et al (2006), Y. Zhi et al (2013), M. Thony et al (2016), and J. She et al (2017) contributed with their method in merging vector and elevation data to create primitive road displays without width and texture but lacked the ability to generate and display intersections. Yi Li et al (2019) 3D scenes were based on image rendering, and Y. Wu et al (2015) defined large-scale multi-view 3D scenes by combining 2D road panoramic maps, satellite map textures, and 3D city models, but their roads lacked height, limiting their ability to show slope changes and did not generate road network topology data for traffic simulation. The last author Bruneton et al (2008) enhanced elevation data to generate roads with fluctuating slopes but there were still disparities in lane textures between virtual scene and the real world, and the supported types of intersections were also limited.

After reviewing the numerous studies, the author claimed that even if current methods can produce 3D scenes, yet automating road details, especially in mountainous regions and various intersections during the generation of large-scale scenes, they remain challenging.

Going to the second part of the problematic that is how to model traffic networks, the current traffic network simulation model was made by Q.I. Yang et al (1996) that primarily use nodes, link, segment and lane to describe the semantics of traffic networks. Wilkie et al (2012) introduced the arc representation method for lane vector data, it allowed us into converting low-detailed GIS traffic network data into high detailed data. There’s also Mao et al (2015) who succeeded in preserving details while improving the efficiency of obtaining vehicle location information during motion simulation, their method consisted of using compressed point columns to represent lane. Or Wang et al (2014) who presented a heretical traffic network semantic model for vehicle group animation simulation, automating the generation of route conflict relations and reducing data input.

Nevertheless, all the existing models demonstrate a strong connection between lanes, requiring laborious recalculation of topological relationships if lane directions change. Additionally, the input of intersection data is challenging and cannot be directly obtained from existing GIS data.

Taking inspiration from the previous models, the paper introduces a method executed in three steps:

- ✓ Define the semantic structure of 2D road network based on the road center axis data.
- ✓ Generate the 2D road shape data by segmenting the elevation data to obtain the elevation data for road area.
- ✓ 0Smoothing the elevation data of the road area, resulting in the achievement of flat smooth 3D road surfaces.

After the method has been designed and implemented, the system proposes to verify the efficiency that this solution provides. It showed a great advantage in generation of large-scale 3D road networks, and the difference was apparent when it meant modeling intersections. When other methods had only X-type[2] or T-type[3] intersections, or modeled only by rectangles with narrow triangles, without forgetting the lack of details that each method had difficulties to attain (wither reconstruction of texture or get smooth curves. As for the paper’s methods, it had detailed features that it managed to portray the longitude difference between country roads and urban areas that are often hard to model, and by that achieve a smoothness that none of the methods could achieve. Not only that, but this method also demonstrated that it could reconstruct 3D road scenes, taking into account different types of roads, intersections, mountains and ramps, with a degree of consistency and connectivity between segments, giving actual road specifications.

Although, the method proposed had various advantages and covered the lack that other methods couldn’t acquire, but there are challenges that this method was also limited. Firstly, due to the lack in input data, the method ignores bridges, overpasses and tunnels. Roundabouts' definitions are not included in the 2D road shape data, and finally roads get stretched because of the altitude difference between road area and non-road area which can lead to a slight inaccuracy.

---

[2] Refers to a road junction where two roads intersect each other at a perpendicular angle resembling the shape of the letter “X” or a cross.

[3] One road meets another at a perpendicular angle, forming a “T” shape.

IV. Recent GIS disaster management systems:

To contribute to the advancement of disaster management system development and to enhance our understanding in the field, we made the decision to incorporate three additional papers into our studies. These papers were chosen based on the potential to provide valuable insight and complement our research objectives. By integrating these sources, we aimed to enrich our understanding of the current methodologies, and strategies employed in GIS disaster management systems.

- Fusion of sensor, AI technologies with GIS for disaster management

Kemper et al (2020) [4] discussed the significant importance of modern technologies in Disaster Risk Management (DRM). These technologies play a crucial role in facilitating the mitigation, preparedness, response, and recovery phases of disaster management. Starting their research with GIS, mentioning how they revolutionized data capture, management, analysis, and presentation when it means disaster risk management. Numerous research works has focused on using GIS for DRM with additional participative approaches such as community-based systems with integration of sensors.

Transitioning with the integration of both GIS and remote sensing (RS) to introducing all types of sensors such as:

- **Satellite Images:** With improved spatial resolution and multi-spectral capabilities, they can offer a cost-effective alternative of aerial images. They can also provide crucial information for post disaster response, risk and preparatory assessment, providing valuable insights for response teams.
- **Aerial Nadir Images**: offer high-resolution and accurate spatial data for mapping and geodata management. This imagery can be useful in disaster monitoring with the advancement in camera technology to enhance the accuracy and range of aerial imaging applications.
- **Oblique Images**: Captures detailed 3D city models with less accuracy than Aerial images (Figure 8), but they still are increasingly utilized in disaster assessment and urban planning, especially when we need to get insights into the degree of damage to buildings. They can also be accessible immediately after capture and can be swiftly processed for GIS-based assessments.

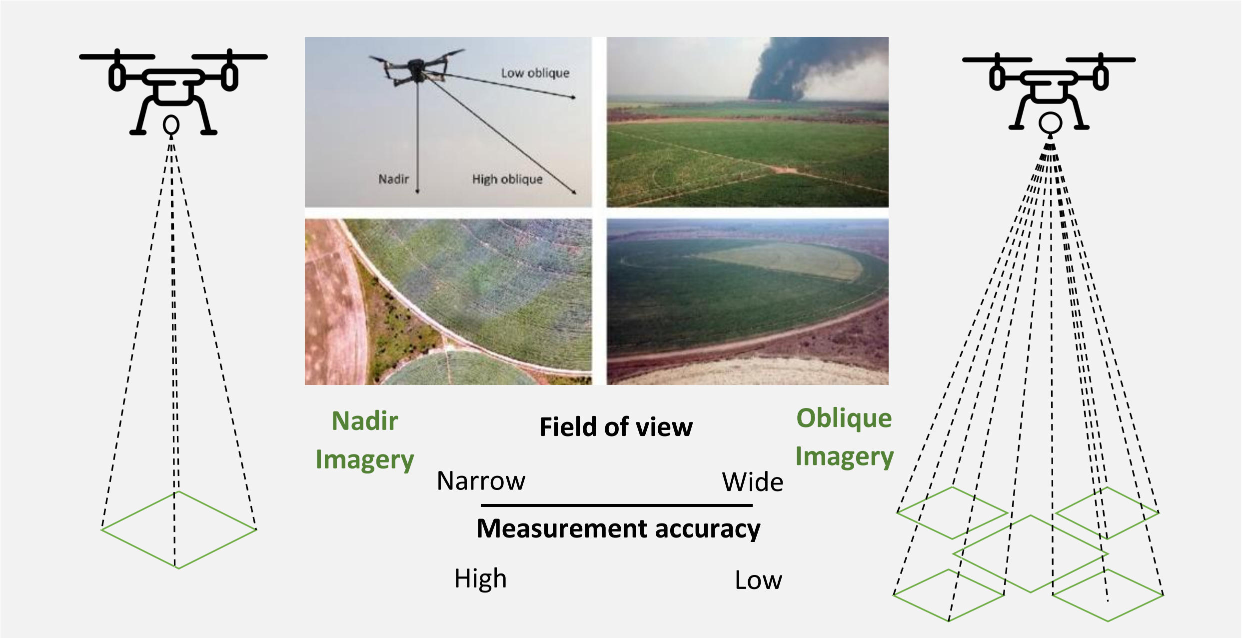


Figure 8: Difference between Nadir imagery and oblique imagery [16]

- **LiDAR:** Generates dense digital surface models (DSMs) and digital terrain models (DTMs) with exceptional precision and detail. They can play a significant role in disaster management through detecting alterations both before and after catastrophic events.
- **Radar and Hyperspectral Satellite Data:** Complement satellite sensors due to their ability to penetrate clouds and observe environmental and geological formations. Despite its inferior resolution compared to LiDAR, they can effectively detect geological shifts and alterations in terrain post natural disasters only.
- **Thermal Cameras**: Detects fluctuations in temperature, used for environmental monitoring and identifying of risks such as damaged power lines and gas leaks. By integrating RGB imagery with these cameras it will facilitate the evaluation of structural damages in buildings.

This integration of various recent technologies aims to streamline data acquisition, making it more convenient and effective for various disaster management phases including prediction, mitigation, response and recovery. The author suggests that merging multiple sensors with AI technologies within GIS software applications can offer highly effective solutions for risk management systems in the future. This approach can be very promising by providing real-time data insight and facilitating the decision-making process.

- Integration of AI to GIS disaster management systems:

Abid (2021) [7] conducted an extensive review of research papers after the identification and distribution according to several aspects, only recent and pertinent studies were selected. The findings were that artificial intelligence had a crucial role in disaster management, encompassing computer systems ability to simulate human behavior and enabling intelligent decision-making. They also mentioned a subset of artificial intelligence that is Machine Learning (ML), which allows systems to learn from data patterns without explicit instructions, facilitating tasks like change detection on satellite images or predicting disaster-affected areas.

AI aims to improve the efficiency of disaster management by enhancing information exchange that can be done by analyzing social media. They can be considered as valuable information source during disasters, using AI techniques to provide insights into the impacts and aid in decision-making, disseminating real-time updates and coordinating relief efforts. Technologies such as robotics, drones, machine learning, deep learning, sensors, and algorithms have evolved, facilitating prompt and efficient decision-making, emphasizing the significance of structured procedures to reduce the negative effects of disasters and acting as a multiplier for safeguarding lives and property.

The author dedicated the following sections to geographic information systems (GIS) and how they serve as essential tools in organizing and visually presenting critical information during emergencies. Geospatial analysis has never been effortless without GIS as it provides graphical outputs such as maps, graphs, and tables for research and forecasting to evaluate flood warnings and develop mitigation strategies. As for early warning systems, all it needs is to incorporate remote sensing and satellite imagery, enabling effective disaster preparedness operations and aid in monitoring disaster zones. By providing data in layered formats, GIS offers comprehensive visuals of hazard zones, including flood-prone areas, buildings, infrastructure, and vulnerable communities. Remote sensing techniques complement GIS by monitoring flood activity and assessing damage before and after disasters through high-resolution imagery. Overall, GIS serves as a robust tool for spatial analysis and plays a vital role in enhancing disaster management practices.

The utilization of advanced AI systems and geospatial technology that GIS provide can improve the planning of crisis response by considering various factors like morphology, weather conditions, and resource availability. It enables the spatial examination of flood risk and can be effective in many other applications to enhances preparedness and crisis management.

- Urban disaster management systems with BIM and GIS:

Cao et al (2023) [5] conducted a comprehensive review of studies to explore the applications of building information modeling (BIM) and geographic information system (GIS) integration in disaster management, identifying its effectiveness across three key stages: disaster prevention and mitigation, disaster response, and post-disaster recovery.

- **Disaster Prevention and Mitigation:** BIM-GIS integration employs various methods like GIS-remote sensing integration and multicriteria analysis, utilizing historical disaster data. In flood management, it enables scenario simulation to predict damage, affected areas, and flood-prone zones using technologies like UAV photography and hydrodynamic modeling. Vulnerability assessments combine BIM, GIS, and computational engines to evaluate urban infrastructure resilience. At the building level, it aids in detailed flood damage assessment, including data preparation, damage evaluation, and economic loss estimation. To improve efficiency, data integration approaches between BIM and GIS are proposed for better disaster management.

- **Disaster Response:** BIM-GIS fusion has proven to be highly effective in the prevention and mitigation phases, allowing for better preparedness and resource allocation. It improves disaster detection by providing real-time data and visualization, and supports emergency evacuation and rescue operations by offering, up-to-date information on building conditions, accessibility and safe pathways, which will be discussed in more details in the following sections:
  - **Disaster Detection and Warning**: BIM-GIS enables precise disaster positioning and 3D demonstrations at micro and macro levels. Risk management systems utilize this integration for real-time monitoring of areas prone to fires, explosions, with machine learning aiding in hazard assessment. The remote sensing and Persistent Scatterer Interferometric Synthetic Aperture Radar (PSInSAR)[4] technologies merge allows early detection of geological hazards. Flood risk systems benefit from BIM-GIS by managing geospatial data and simulating flood-prone areas.
  - **Emergency Evacuation and Rescue:** BIM-GIS supports personnel positioning and route planning during emergencies. Indoor monitoring systems locate trapped individuals, while route planning considers geometric elements and structural conditions for safe evacuation. Integration with Smart City initiatives enhances facility management and enables efficient disaster rescue operations.

- **In post-disaster recovery:** The integration of BIM and GIS is pivotal for evaluating damage, assessing reconstruction needs, and expediting management processes. Simulators accurately assesses post-earthquake building conditions, while predictive frameworks estimate renovation costs using factors like drift degree and acceleration. Remote sensing technologies aid in assessing historic building damage and bridge conditions. Streamlining reconstruction processes, a BIM-based Virtual Permitting Framework[5] (VPF) automates approvals and matches repair schemes, while smart management systems improve efficiency by facilitating search and management of building repairs. Tilt photography supports transportation planning Integrating BIM and GIS in historic building renovation provides knowledge management for retrofitting efforts. Overall, BIM-GIS integration enables a comprehensive approach to post-disaster recovery, from damage assessment to reconstruction management and historic building renovation.

The paper explores the potential benefits and challenges of BIM and GIS for urban disaster management. One significant challenge is the compatibility issue: merging data from BIM

---

[4] A remote sensing technique used in geospatial analysis to measure ground deformation with high precision.

[5] Framework that facilitates the management of documentation, compliance checks, and coordination among multiple agencies, improving efficiency, reducing errors, and expediting the overall permitting process.

and GIS systems faces hurdles due to different formats and standards. Additionally, managing diverse urban environments requires coordinating various data sources and sensors, making data collection more complex. Furthermore, current research tends to focus on specific types of disasters, leaving gaps in preparedness for others like snowstorms and hailstorms. Overcoming these challenges necessitates standardized data exchange protocols, creative data collection approaches, and broader research coverage of urban disasters. Nevertheless, the study highlights the importance of overcoming these hurdles to fully leverage BIM-GIS integration for enhancing urban disaster management strategies.

V. Overview of existing research

After a comprehensive review, it became apparent that methodologies vary widely among studies. While most research papers employed quantitative analysis, utilizing multiple studies to assert concepts, only a few opted for a qualitative approach, devising their own solutions to evaluate efficacy. Among the studies examined, the first study [2] proved to be the most pertinent, providing valuable insights into the integration of three key technologies and their potential impact on disaster management systems. It offered a clear understanding of how this integration could enhance coverage across all phases of disaster management, emphasizing the importance of leveraging technological advancements such as integration with existing traditional systems to address urgent challenges presented by natural disasters.

The second most accurate research [3], despite the absence of implementation of edge computing in their proposed system compared to the first study, the author gave a special importance to elevation data when developing a three-dimensional GIS. They asserted that such emphasis could enhance map quality, rendering it closer to reality. This result emerged as a response to the challenge of creating a large-scale map while managing substantial geospatial and graphical data volumes. Attempting to process such data on a single desktop PC led to the neglect or slight alteration of some road curves. Additionally, the system proposed did not include representations for bridges at that time.

The last section [1, 4, 5, 7] comprised recent research studies that explored the integration of recent technologies to a GI Systems. Notably, all papers shared a commonality in the absence of both virtual reality and edge computing. To summarize their findings, we reference studies [7, 1], which underscored the significance of remote sensing (RS) and recommended the utilization of powerful sensors for more efficient and rapid results. Additionally, the integration of secondary technologies into GIS was deliberated in both studies: [7] which introduced artificial intelligence (AI) to RS and GIS, and [4] preferred putting more focus on the integration of artificial intelligence and the promising path that it offers if it gets merged with GI systems. Moreover, [5] discussed the integration of building information modeling (BIM) with GI systems. Through a systematic review, the paper delineated the capabilities of this technology but concluded that further research in data acquisition methods is imperative.

Before proposing our solution, we must address one last aspect in the previously cited papers. While we reviewed the chosen methods, the results, and the challenges they faced in presenting their studies or systems by merging different recent technologies, we didn't emphasize the implications of their research on disaster management. Thus far, except for

[3] Which focused more on obtaining a large-scale map and improving representation and data management, all the previous papers shared a common goal: to enhance efficiency in disaster management and cover all phases of disasters, where time frequently plays a crucial role in decision-making and response effectiveness. This underscores the significance of integration efforts.

VI. Conclusion

After examining the contributions of each study, it's evident that the integration shows promise for enhancing disaster management. However, not all proposals successfully address the research question of creating a comprehensive system to mitigate catastrophic events and reduce negative impacts.

Common among them is an emphasis on GIS representation, often utilizing elevation data and high-resolution satellite imagery to enhance realism and accuracy. Nevertheless, achieving detailed results necessitates loading small portions of datasets, a challenge that only edge computing can address by enhancing system efficiency.

Additionally, recent GIS disaster management systems also presented favorable solutions, yet some rely on energy-intensive equipment and data processing, posing challenges for implementation in larger geographical areas.

Building on these insights, the following chapter will explore the chosen methodology, prioritizing user interaction, scalability, and adaptability. The goal is to bridge the identified research gap and ultimately enhance the effectiveness of disaster management.

Chapter four: Methodology

I. Introduction

This chapter provides a comprehensive description on the methodology employed in the research. By evaluating the studies and proposed systems that were previously outlined in chapter three, it will delineate the principal requirements that the solution must satisfy in order to achieve a more optimized and seamless application.

The following section present contextualization of the actors and the scenario of a traditional disaster management system, based on the ORSEC plan (plan organisation des secours) or basic urgent plans in Algeria. Once the actors and their responsibilities have been identified, the process of establishing sequence diagram, and case diagrams becomes a straightforward process.

At the juncture, it is necessary to illustrate the solution's architectural framework, which summarize the scenario's progression, the actors' information exchange, and the efficient management of the disaster. Subsequently, the chapter outlines various stages that the solution took to get developed. Divided into two parts, from a front-end point of view, we must define a geographical information system that will result in a web map, which is characterized by layers of geodata and satellite imagery. The virtual map must then follow realistic standards to be considered as virtual reality. From a back-end perspective, the edge computing concept is introduced, whereby datasets are split according to the actor, each actor (end-user) owns a dataset with geodata and information, even maps that they only need, to ensure the exchange of data between actors, a communication system must be established.

II. Contextualization:

This segment will serve as a bridge between theoretical bases we previously set and the design phase that our solution will take. Foremost, we need to acquire more knowledge on the disaster management field of our country, and based on numerous experiences throughout the years, these are the major disasters that represent the highest risk to happened:

- Earthquakes and geological hazards.
- Floods.
- Climatic hazards.
- Forest fires.
- Industrial and energy risks.
- Radiological and nuclear risks.
- Risks to human health.

The scenario will take place in accordance of the ORSEC PLAN [21], an "*emergency response plan defined as an organizational mechanism that complies with the regulations in force. It is also an essential tool for decentralized authorities in disaster management and the care of disaster victims. Additionally, it is important to note that no single organization can deal with a disaster on its own; it involves all the public or private structures that can contribute to containing the damage*".[18]

These decentralized authorities can be defined by a range of services such as police, SAMU (Service d'Aide Médicale d'Urgence), fire department, gendarmerie, equipment, and more. But the most important authority must be Civil defense, "*a set of civil measures and provisions designed to guarantee the continuity of national life in all circumstances*".

These services can be summarized within the following table:

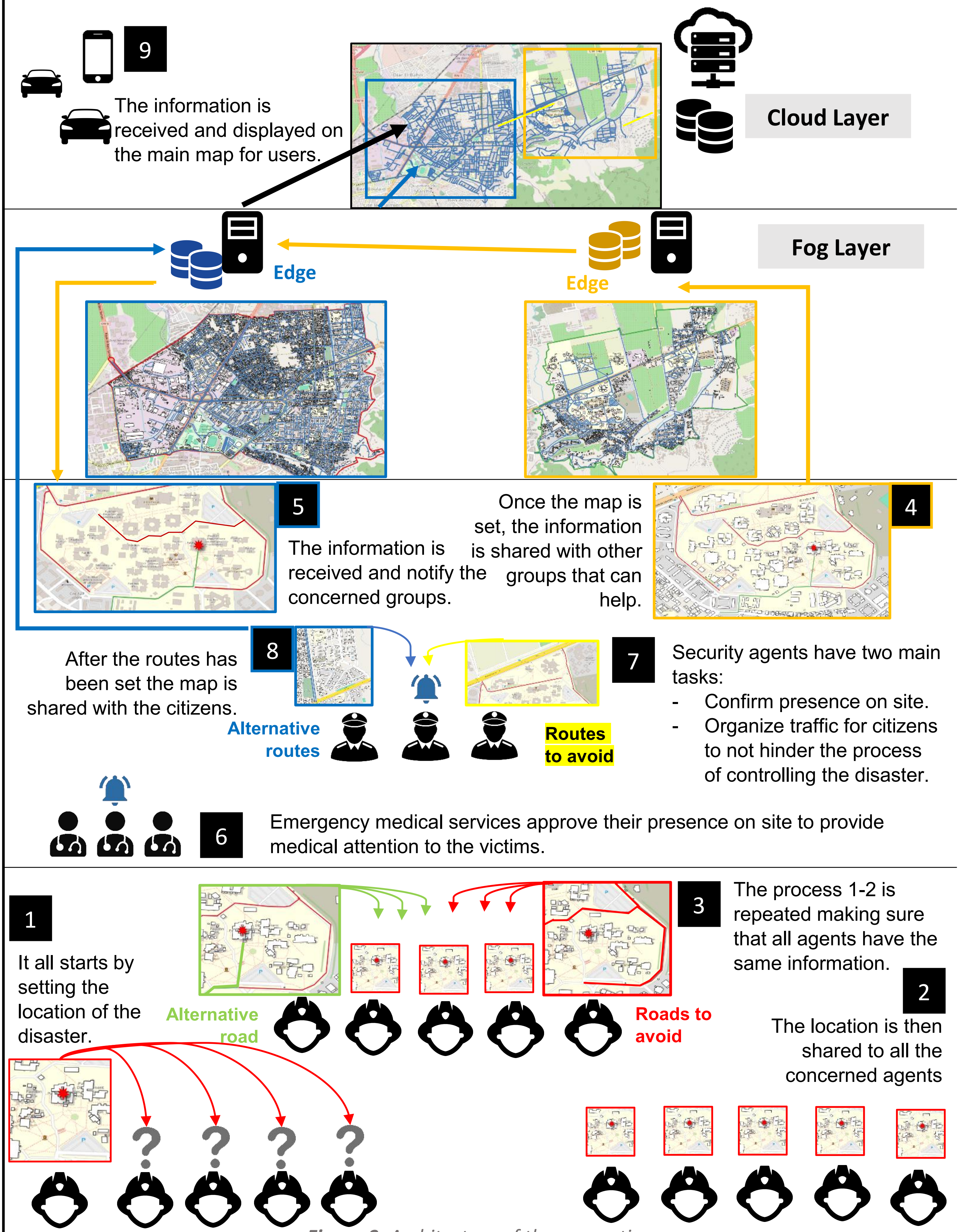
Table 1 *: Main actors in a Disaster Management scenario according to ORSEC plan or any other emergency plan in Algeria [19],[20]*

Actor	**Actions**
- Civil protection	- Prevent/limit all consequences. - Protect property and provide humanitarian services. - Creation of new structures, to reduce vulnerability to security breaches of sensitive installations, communications and access to vital centers. - Assure all search, rescue and recovery operations.
- The national gendarmerie - Security of Daira	- Participate in the defense of the fundamental interests of the State. - Ensuring the safety of citizens. - Maintain public order. - Organize assistance for disaster victims. - identify injured and deceased individuals. - Preserve and recover archives. - Organize vehicle traffic.
- Emergency medical service	- Stabilization of vital distress. - Diagnosis. - Emergency decontamination. - Psychological care of disaster victims. - Monitoring progress under treatment. - Thorough decontamination. Here, medical and surgical emergencies take precedence over radiological emergencies. - Monitoring the healthiness of premises.

III. Communication [20] must be distributed throughout the danger zone and meet the following requirements:

- Quickly visualize the location of various zones.
- Allow control of the movements of personnel and other people involved.
- Facilitate communication between services, people involved and victims.

It consists of limiting the risks incurred by the civilian population and reducing the damage caused to material resources and wealth of all kinds, by organizing assistance in the event of accidents, disasters or cataclysms. In conclusion, (Figure 9) must be added to illustrate the solution's architectural framework, how the actors previously stated in contextualization section act and communicate with each other in a disaster scenario.



The information is received and displayed on the main map for users.

The information is received and notify the concerned groups.

Once the map is set, the information is shared with other groups that can help.

After the routes has been set the map is shared with the citizens.

Security agents have two main tasks:

- Confirm presence on site.
- Organize traffic for citizens to not hinder the process of controlling the disaster.

Emergency medical services approve their presence on site to provide medical attention to the victims.

It all starts by setting the location of the disaster.

The process 1-2 is repeated making sure that all agents have the same information.

The location is then shared to all the concerned agents

Figure 9: *Architecture of the conception*

IV. Requirements

This section aims to establish the specific requirements of the system, including data needs, functionalities, and user interface design. Followed by the implementation details, focusing on data management, selection of appropriate GIS software and web development tools, and system architecture design. Finally, it will result in in the creation of a user-friendly and informative web map, allowing for visualization, analysis, and interaction with spatial data relevant to disaster management system.

- Geographic information system development

Developing a GIS for advanced disaster management system involves a robust technical infrastructure, and a comprehensive understanding of spatial data. This subsection outlines the various requirements to accomplish in order to create a web map, focusing to reach the highest resolution and accurate representation, to be considered as virtual reality according to the definitions and theories that have been previously delivered. Going through the phases that guided the map implementation, it includes:

- **Data acquisition :** The initial phase of the development is critical. It involves partitioning the map into layers, commencing with the first layer, which consists of creating a vectorial map. This process entails representing the map with simple geometric shapes: roads are modelled as polylines, locations as points, and buildings or zones, such as greens or landscapes, represented by polygons.

  In this case, points are not required since the user will create them to emphasize the disaster's location. Furthermore, it is unlikely that geospatial datasets will be obtained, as they are considered confidential data used only for enterprise, education, or military purposes. Therefore, roads were created manually to simplify the representation and facilitate distinction.

  Moreover, for the buildings, an open-source dataset [26] of building footprints was provided by Google Open Buildings. Their datasets are represented within a grid, allowing the user to download only the buildings within the area of interest. This approach bypasses the necessity of loading the entire database, which can be unwieldy to display, before deleting the superfluous footprints.

  It is important to note that this data collection is a simple geometry representation and lacks of information like building height, locations and occupancy, so further modification to fill the gaps according to the exigency will be needed.

  QGIS has been used to construct the web map, an Open-Source Geographic Information System (GIS) software licensed under the GNU[6] General Public License. Every plugin, tool and functionality can be defined by a python function making it easy to formalize according to desire following the documentation.

There are two essential keywords that compose a web map that should be defined to make the process more comprehensive:

---

[6] GNU's Not Unix, in another way it doesn't contain any Unix code.

- **Base map**: serves as the foundational layer for the map, it provides the background context to other geographic data. It also establishes the spatial reference for the data, giving better understanding to location and relationship between different geographic elements.
- **Overlay maps:** are the thematic layers, spatial data that were previously mentioned which are point data, polyline or line data and polygon data. Overlay maps are responsible of providing data that can be visualized and analyzed.

As a start, the foundation map that has been employed to create roads and overlay buildings on was an Open Street Map, then a “Clip all layers” plugin was executed to get rid of all building footprints that fell outside the area. The process is shown within (Figure 11) where the area has been defined with one polygon layer.

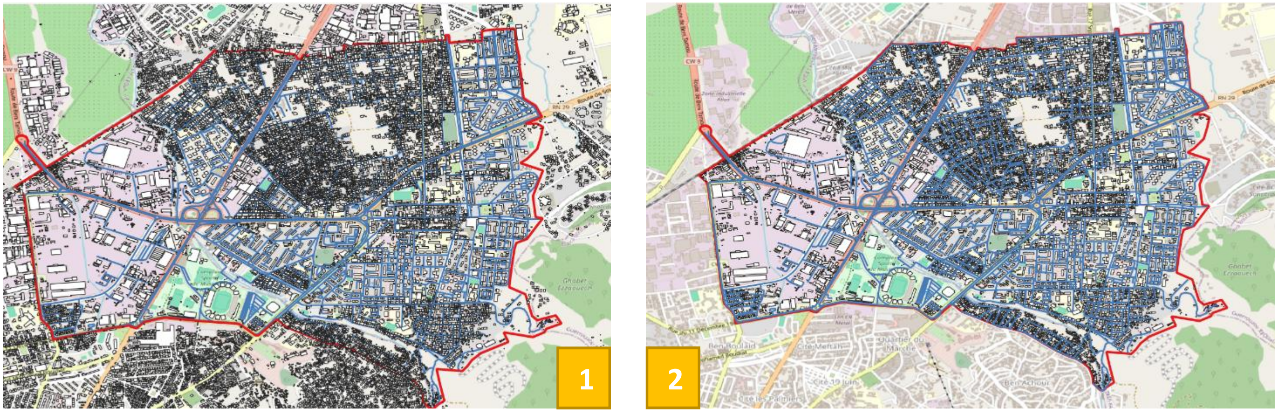


Figure 10: *Process of loading and cleaning the map from unnecessary building footprints for both edges’ map.*

1: Data has been loaded

2: Clip all overlayed layers within the red polygon.

- **Visualization and User Interface**: Now that the data acquisition has been done, this section will include creating and developing 3D visualizations and export the final result to be implemented in the web application. To do the transition it requires first changing the base map given that it’s been utilized to create the vectorial data, and replace it with satellite imagery. It was done by laying at first a satellite imagery base map, then export new preprint to select only the wanted area and save it in the highest resolution and the first reflections that can be made are :

- In terms of contrast, the colors are more vibrant and some area gained more texture that can be distinguished even with the vectorial layers laid on top of the base map.

- It allows not only to get better render but to set the zooming scale to the maximum without losing credibility in the representation.

Noting that the main difference between the two pictures presented in (Figure 12) is not just the resolution, but also the area surface. (1) is base map that have coverage worldwide, when the second one that got exported covers only the edge’s map that’s why it was possible to get high definition without worrying about the storage or the latency that it was about to take to get loaded on the web application.

Another key concepts that should be mentioned when talking about satellite imagery and three-dimensional visualization are Raster data, Digital Elevation Model data and Hill shade.

- **Raster data** [27]: Contrary to vector data, Raster analysis is a grided images constructed out of pixels or cells, each pixel holding a value that is utilized to portray different characteristic of the map such as occupancy, temperature or elevation. Its resolution can be determined by the size of the cell, the smaller the pixel is the finer the detail it will provide. Additionally, Raster data can store various data types such as integer for elevation models, or floating-point for temperature in degrees Celsius.
- **Digital Elevation Model** (DEM): is a type of Raster data that define the elevation of a terrain surface, each pixel in a DEM model holds a value that corresponds to the height above a specific reference level for example sea level, but also can hold minus values that will represent depth that can be noticed in rivers, seas or any landscape in general. It also has many types and raster calculations that can be made to add a more characteristics to the map like Hillshade that simulates illumination of terrain to create highlight, or Slope analysis which is a valuable parameter that can calculate the classification, landslide susceptibility or habitat suitability modeling and many other available calculations and types of elevation models.

To clarify these concepts, different calculations has been implemented to one of the maps and illustrated within (Figure 11)

(1): Digital Elevation Model are sometimes tricky to acquire since most of the providers are not open source. But for this analysis, an API key was provided from Open Topography [28] plugin that requires a free subscription. Furthermore, vectorial data has been laid on top of the Raster DEM layer to show how it calculated the layer only around the wanted area and no clipping process is needed to be executed again.

This is a Hillshade layer (2), this type of DEM analysis along with other types doesn't require any data collection since they get calculated from the elevation model itself.

As for the last picture (3), it's the result of generating a 3D view by merging the previous satellite imagery base map with the elevation model, along with the building layer disabled to have a better view on the landscape we notice how the terrain isn't flat anymore and how by adding height it manages to get better visualization.

Figure 11: *Different Digital Elevation models' analysis and how it can improve visualizations and representation of a satellite imagery base map.*

While this perspective holds merit, it is important to consider that the goal of the application is to serve as a disaster management system that assure geospatial data and adding Z coordinates that represents height might integrate only more information to communicate. In a response system where the main requirement is communication and doesn't need to account for Raster data. Besides, the path that should be taken is opting of a user-friendly interface that will make the illustration of any situation effortless.

Going from that perspective, the next step would be to add height to the buildings to obtain what falls between a 2D and 3D representation called 2.5-dimensional representation. Despite the flat terrain representation, the building layer will add that touch of depth and additional factors. And as mentioned in the GIS development section, building footprints had gaps in terms of data since they were only a geometry representation. While the locations got resolved by merging the layer with the base map that had geospatial data defined, there's only the height that's left. And to solve this gap there's two possible solutions that can be implemented illustrated in (Figure 12):

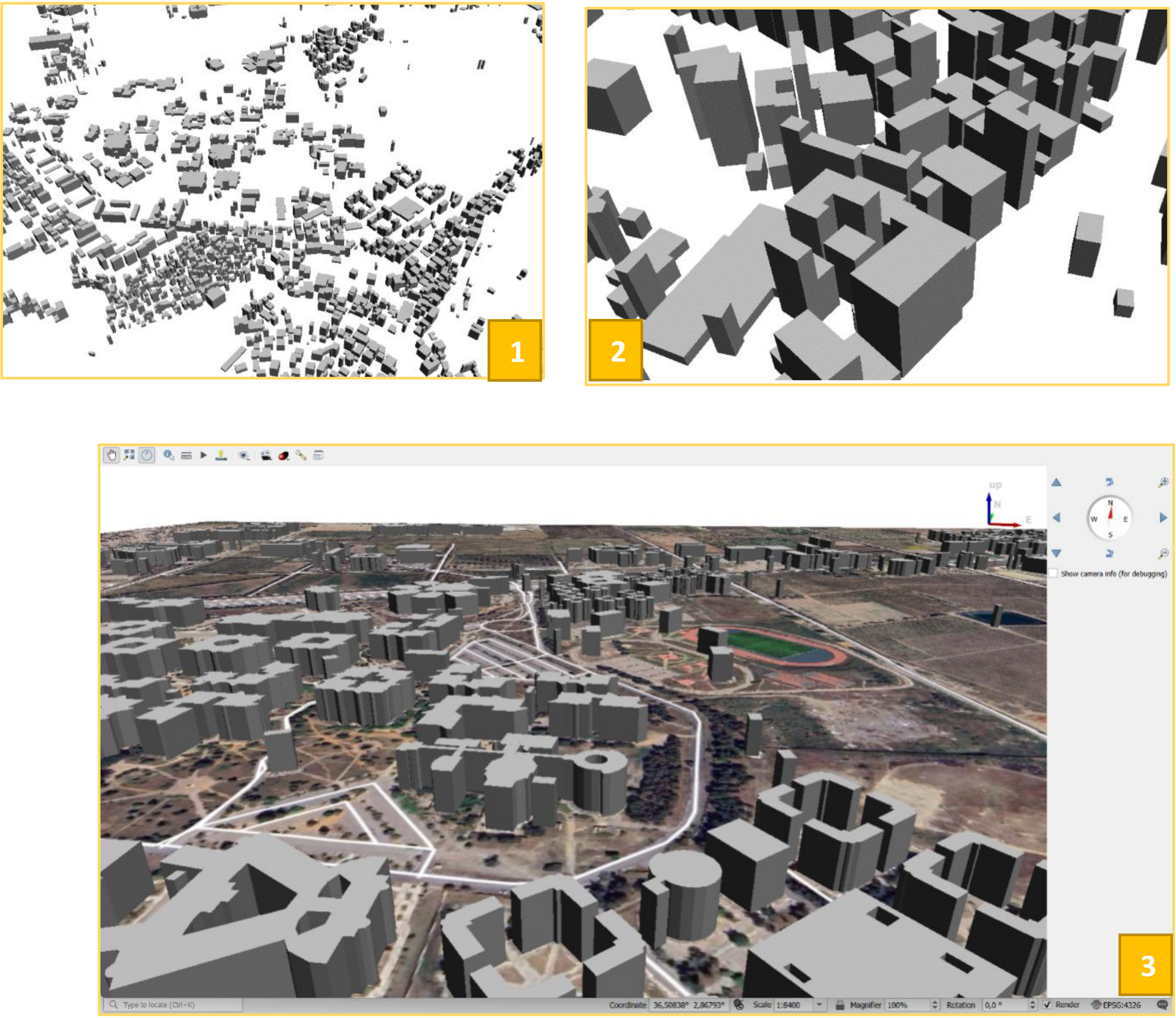


Figure 12: *Two different solutions to solve the lack of height data in building footprints.*

The first solution shown in (1,2) of the (figure 12) was obtained by adding the height as a field in the attribute table, then executing the Random () function with values going from 3 to 26 to symbolize the average height of the buildings located in the area of interest in this case city of Blida in Algeria. But this solution as much as it fulfilled the requirement but in terms of accuracy, it's noticed that it lacks of realism when buildings of the same neighborhood with the same structure get assigned with different height values.

That's why the second solution came to resolve this problem, other than creating a new field in this case, an extrusion method has been employed, extruding the buildings by Z value uniformly.

As a final result, and to summarize the user's interface, or in another terms, the maps to include and export to the web application, the representation that is seen suitable to exploit and use as a base for overlaying data related to disasters and share them with other users is the satellite imagery base map with vectorial data layers of roads and buildings. As for the display after the buildings have been extruded, it will be left as an additional map to leave to the user the possibility to switch representations according to the desire.

Importantly, during this whole section it has been delivered steps on how to develop only one partition of the map but how about the whole map, how can the user view the map outside their area. If the user's area map is considered as the edge map, then it will be logical to consider the world map as the cloud side map, more like the guest map, doesn't require any vectorial data or satellite imagery because its purpose is to be light weighted in terms of dataset and the user holds only specific data collection that concerns him. Thus, the cloud map has been added as an overlay map contributed by Open Street Map, a simple 2D map that includes roads, locations, and its render is similar to any Google street map. The (Figure 13) summarize the whole paragraph adding the possibility that the cloud map can be disabled if it's not indispensable in some cases.

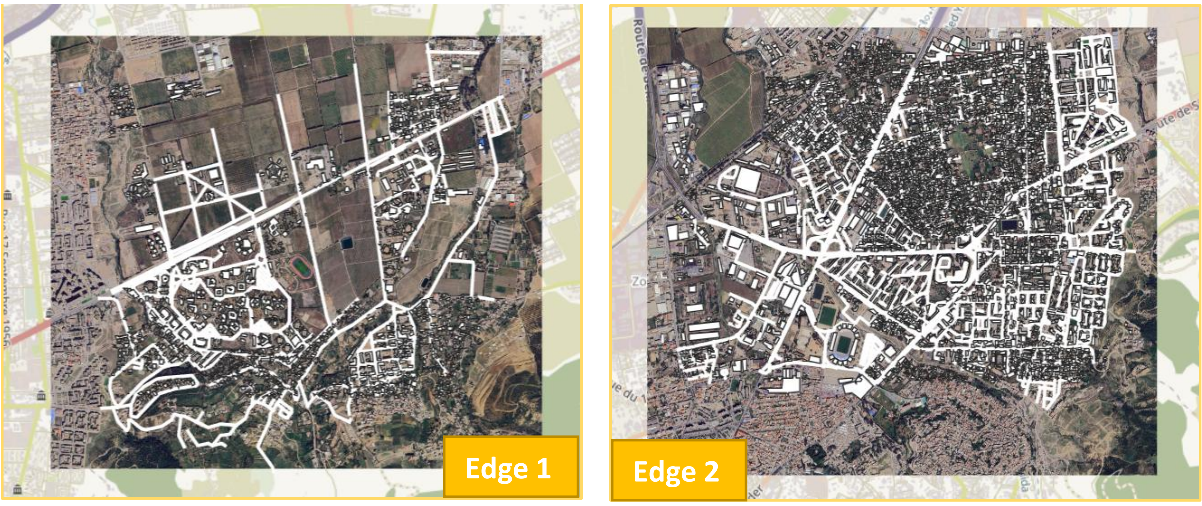


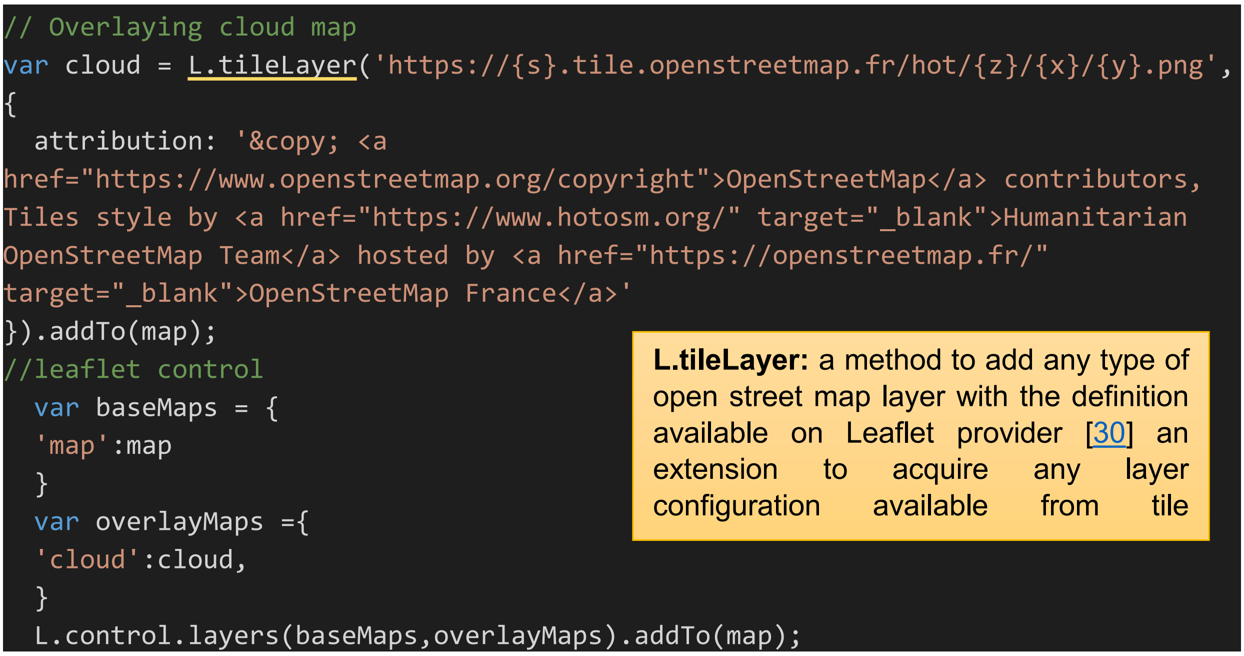
Figure 13: *Conception of the visualization interface by using vectorial, satellite imagery data in two different areas of the city of Blida in Algeria.*

Now that the implementation of map layers with the cloud open street map being the overlay map and the combination of vectorial data and satellite imagery forming the base map. Ensuring the right styling and symbology for the vectorial data collection it enabled the possibility to reach a clear and easy to distinguish details by the user.

- **Interactive elements:** To complete the interface interactive elements should be added, they can be provided by Leaflet [29] an Open-source JavaScript library, widely utilized for the development of interactive maps within web applications. It is meticulously crafted to offer a lightweight, user-friendly, and highly extensible render, making it an appropriate choice to incorporate mapping and spatial data visualization into the project. To define any Leaflet plugin, it is usually composed of two parts:
    - **Definition layer:** Defining the objects that can be added to the map to display geographic data or intercreative features. There are many types of layers such as Tile, marker, or even vectorial layers if specific data is required to be displayed on the map such as line string, circle, polygon or any other GeoJSON data layer.
    - **Layer of Control:** By using the "L.Control" method and considered as the base class for all Leaflet controls. By extending this class it allows to add specific

functionalities and add them to the map employing other methods and options such as “onAdd” and “onRemove” methods to tailor the geospatial data according to the requirements.

To illustrate the previous notions, a piece of code is provided within the following (Figure 14):



```
// Overlaying cloud map
var cloud = L.tileLayer('https://{s}.tile.openstreetmap.fr/hot/{z}/{x}/{y}.png',
{
  attribution: '&copy; <a
href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors,
Tiles style by <a href="https://www.hotosm.org/" target="_blank">Humanitarian
OpenStreetMap Team</a> hosted by <a href="https://openstreetmap.fr/"
target="_blank">OpenStreetMap France</a>'
}).addTo(map);
//leaflet control
  var baseMaps = {
  'map':map
  }
  var overlayMaps ={
  'cloud':cloud,
  }
  L.control.layers(baseMaps,overlayMaps).addTo(map);
```

L.tileLayer: a method to add any type of open street map layer with the definition available on Leaflet provider [30] an extension to acquire any layer configuration available from tile

Figure 14: JavaScript code to overlay maps together using L.tileLayer and L.control.layers methods provided by Leaflet library.

Another extension of Leaflet that was used to create an interactive interface is Leaflet.draw, enables the creation of vectorial data according to the actor, such as the civil protection agent need to delineate zones of danger, disaster locations, and define new safe roads or dangerous roads, so drawn feature like polygons, markers or point feature, and line strings are required. When police officers only need to organize traffic so only line strings are going to be included. The definition is as easy as to attribute to vectorial objects values of “true” and “false to include or exclude them into the toolbar that gets added once the L.control method is executed to incorporate it to the map. Plus, a style icon was integrated to allow to the user to change color, opacity, line type generated to make it more comprehensive to distinguish the safe from the danger surfaces, paths. Another piece of code is provided to illustrate the scripting of the toolbar.

Concluding with the final result (Figure 15) of the interface where the layered map with all its interactive elements is added into an HTML and CSS coded page, introducing the next section that will treat the back-end of the web application. Where every button will get its functionality to assure communication and data flow between the server and database, all while respecting a Cloud and Edge computing architecture.

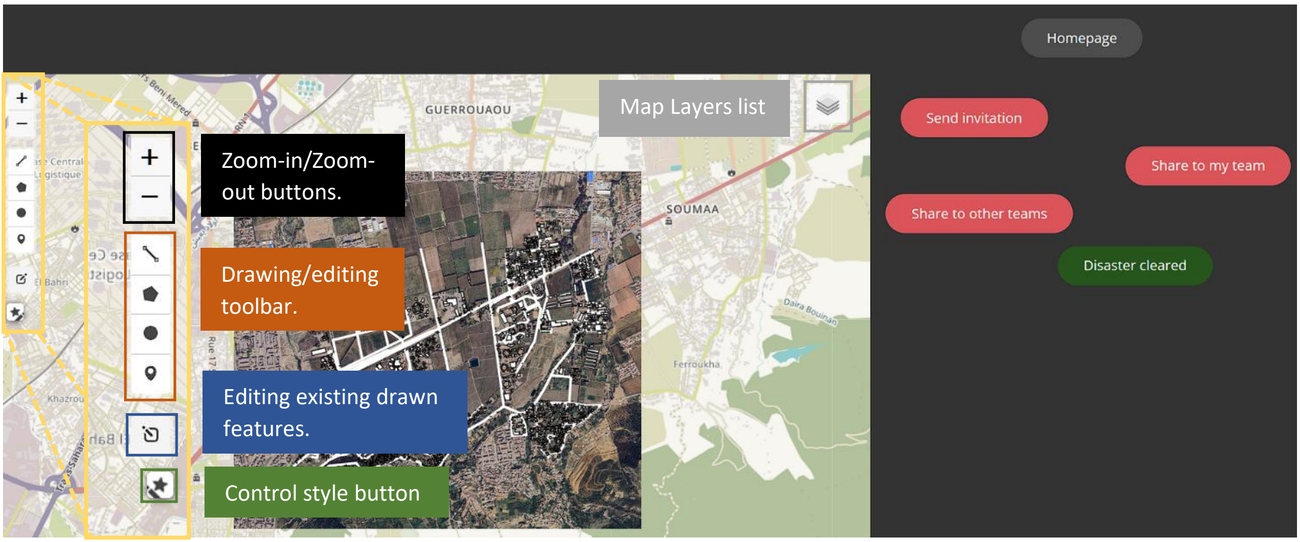


Figure 15: *Example of final render of the interface with all the interactive elements incorporated. (For a Civil Protection agent)*

- Implementation and creation of the web application

The purpose behind developing a web application isn't to present a final product or solution that will be used in real situations, but to set bases for these new technologies to get integrated all in the same environment, studying their behavior and conclude the best application to simulate what can be applied in future.

- **Architecture and database design:** In previous section, an architecture of the solution has been discussed but in a vague detail, prioritizing the disaster scenario and how agents interacted, but this time the architecture showcased in (Figure 17) will simplify the solution using two edges and showing end users' interactions summarized within the following list:
  - **Edge layer:** In this layer the users interact with the server or mainly the database of their Edge that is attributed to the area they are assigned to. Each edge has three types of users assigned to: A Civil protection agent, a police officer agent, and an urgent medical service agent. Each type can have teams and big number of end users each assigned to their function center within the area but under a disaster situation, only concerned users will interact with their local edge server to wither create data, and save it to the database to display it back to all members of the same team.

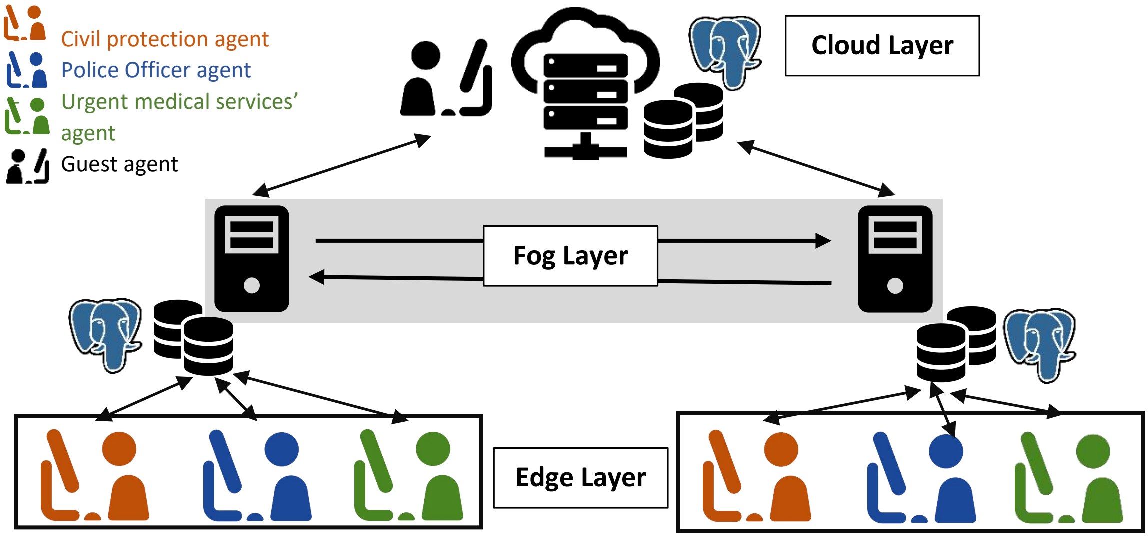


Figure 16: Architecture of the web application's back-end side showcasing client-server interactions.

- **Fog layer:** If the Edge layer describes internal interactions of the end-users with their own database, Fog layer would represent external interactions between edge servers. The purpose of this layer would be to share geospatial data across the area to reach more end-users that can be willing to help controlling the situation or organize the response and disaster recovery.
- **Cloud layer:** This layer is the layer the end users would interact with when geospatial data are needed to be shared with citizens or guest agents. Mostly dedicated for police officers' type of end-users to organize traffic which means display to guest users the roads to avoid in order to not hinder concerned active agents to reach the zone of danger and rescue victims to the closest hospital or care center. Every geospatial data is forwarded to the Edge server that will be directed by then to the Cloud server.

In the development of a web application, database design come to lay the foundation for efficient data storage, retrieval, and management. And to ensures that the database handles spatial data and user communication, PostgreSQL is the one that had been employed in this project. With the "postgis" extension integrated in each database, it enabled various geospatial data types such as: geometry, JSONB, JSON and many other types that may not be available in every SQL database. To simulate the architecture each server has been assigned to a database, and each database was assigned on its turn to a separated table space or storage directory that can be found in the project's directory.

Moreover, to complete the architecture and database subsection and to make the previous user-server interactions more comprehensive, a use case and sequence case diagram were implemented as outlined below:

- **Use case diagram:** Although authorities are included, but we chose the three main actors that acts the most in the disaster management scenario that are: Civil protection, Gendarmerie and SAMU (Urgent medical service) (Figure 17).

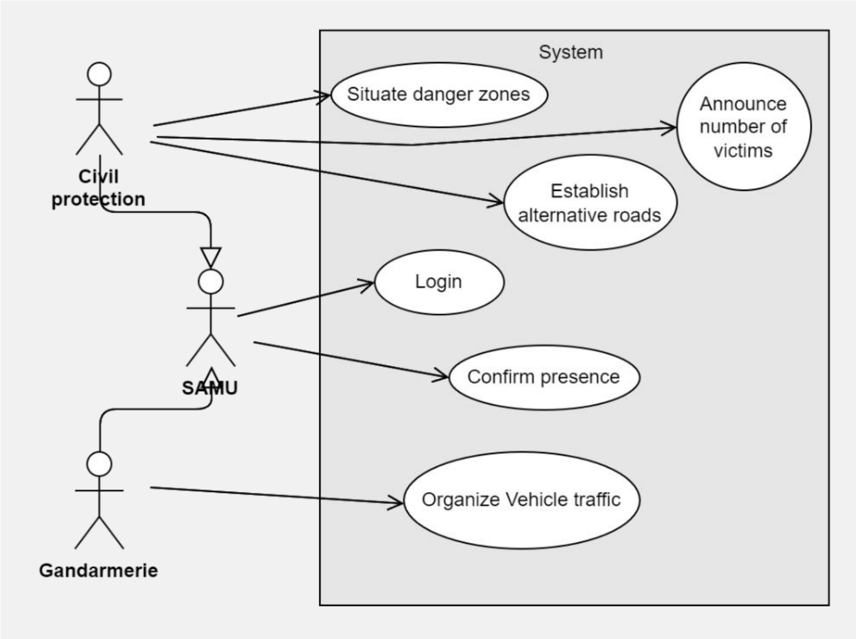


Figure 17: Use case diagram of our application

- **Sequence diagram:**

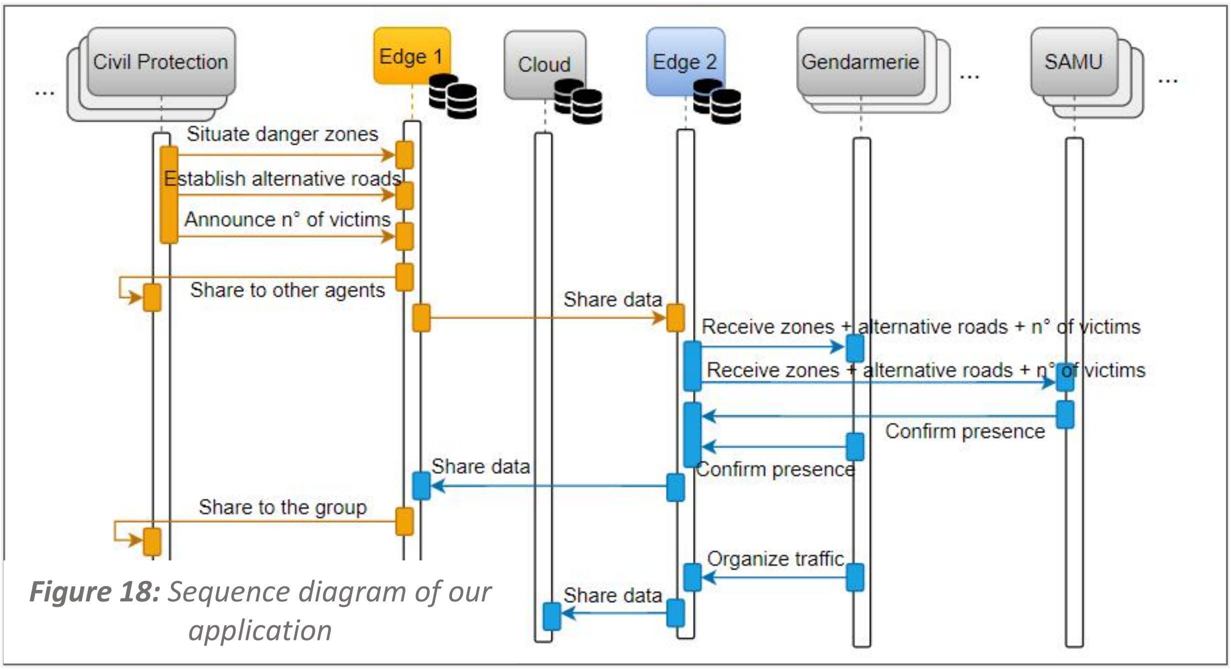


__Figure 18:__ Sequence diagram of our application

- **Authentication:** Dealing with this many layers of servers all in one application change the concept of every usual coding phase that necessitated only one server to handle every permission of all user. But when dealing with users assigned to Edge server the first question that should be asked is “How the login is handled?”, “Where are the user’s credentials stored?” And to answers to those question there’s two possible solutions illustrated in (Figure 19). But before presenting them it is necessary to note that the server that is responsible of the authentication is located on a Cloud level since the user before authenticating, he’s considered as a guest agent, then when the authentication is validated, the user will have the access to the Edge's interface and will interact with their dedicated server.

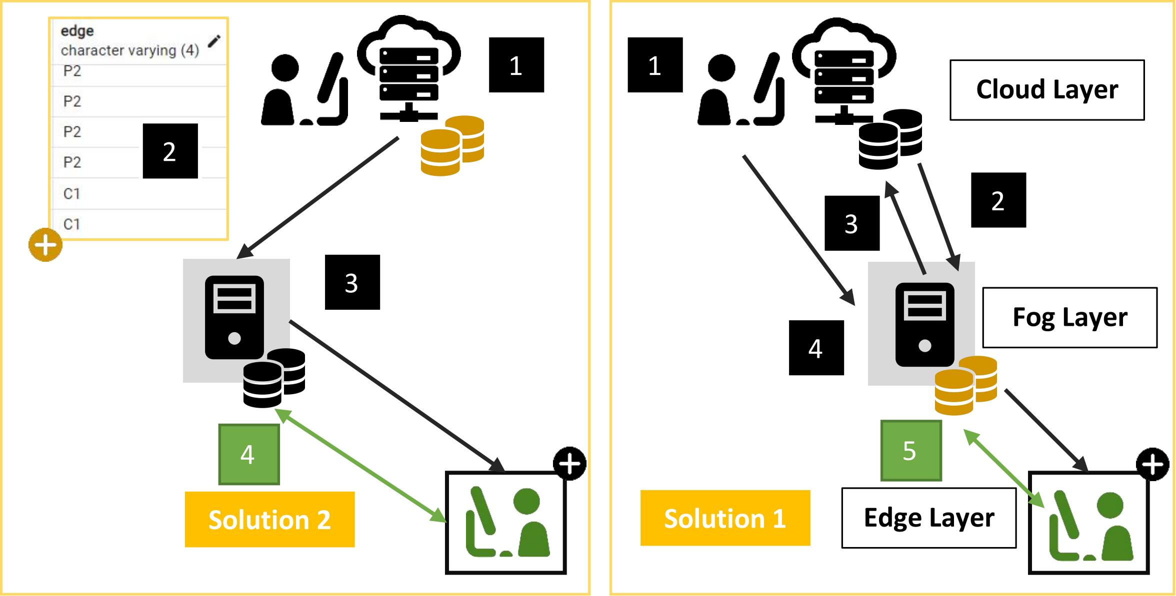


Figure 19: *Two possible solutions for user authentication according to credentials' table location.*

To explain the process of each solution, each numbered interaction should be defined so it will allow a comparison to choose the more adequate solution to the application. For the first solution, credentials table is stored in the Edge's database itself so in order for the user to get authenticated after the introduction of these last (1), the inputs are then fetched to be sent back to the server (2,3) then when it's validated the user get directed to the proper interface (4) so they can interact with the Edge server of their area (5).

As for the second solution, the credentials table is stored back in the Cloud database, once the inputs are validated (1), the server fetch of an extra field added to the credentials table called "edge", a varying character field that specify both the user's Edge and Interface (2) for example to translate some introduced values:

- "P2 = Second Edge to the police interface"
- "C1 = First Edge to the Civil protection interface"

Subsequently, the user will get directed to the proper interface and interact with the Edge server of their area (4). Both solutions offer distinct advantages tailored to different needs and context. The first solution excels in security, storing user credentials locally that can be considered as confidential data. While the second solution provide efficiency, reducing delay that can be caused by the first proposition especially in an environment where latency is essential and where any time-consuming action can lead to critical consequences deducting that the adequate solution that should be applied for this project is indeed the second solution.

V. Conclusion

The main idea of this chapter was to emphasize that the primary issue in most studies when developing their systems is to achieve high resolution rendering that is both accurate and close to reality. By dividing the map by edges, it was easy to produce very high-resolution satellite images. To create a geographic information system, we had to create a vectorial map to characterize the database, then load the satellite images and add fields in the "buildings" layer to extrude them and create a three-dimensional view.

In the contextualization and communication section, it became clear that the main actors to be considered in the development of the system were the Civil Protection, the Emergency Medical Services and the Gendarmerie. It is crucial to point out that the purpose of splitting the dataset is not to create the same dataset for each actor, but rather that each actor has its own confidential data. For example, in the case of civil protection, the map will show risks and information about the terrains. The gendarmerie and emergency medical services, on the other hand, do not need this type of information, so they would only see buildings and roads within their map zone.

Chapter five: Results

I. Introduction

In this final chapter, after acquiring all the necessary knowledge, from defining the three main key concepts of this research to discussing the potential integration to create a seamless solution that would support disaster response and recovery, a comprehensive analysis of the results obtained from the execution of the proposed system is presented.

The analysis begins by showcasing the performance metrics and overall effectiveness, highlighting the achievements and identifying areas for improvement. It delves into the challenges faced during the deployment phase, such as implementing a physical architecture to portray the edge computing and layered structure.

Furthermore, this chapter outlines potential future directions for the system. It explores various possibilities for integrating additional features and concepts that can broaden the system's scope, enabling it to serve larger, more complex purposes beyond disaster recovery. The goal is to envision a path forward where the system evolves to address broader societal needs, leveraging advancements in technology and user experience to create more impactful solutions.

Through a forward-looking perspective, the aim is to inspire continued innovation and development, ensuring the system’s growth and relevance in addressing significant challenges such as disaster management.

II. The adopted architecture and requirements :

Compared to the architecture of the conception, where it brought full clarity to how the solution should be developed. By demonstrating every role of actor, the responsibilities and how the whole process is split into a set of following actions only to make the methodology and perusing results more comprehensive. The architecture of the solution (Figure 20) confirm the notions previously mentioned and display how the solution has been orchestrated according to the available environment of work following the requirements that will be sited in the next sub section.

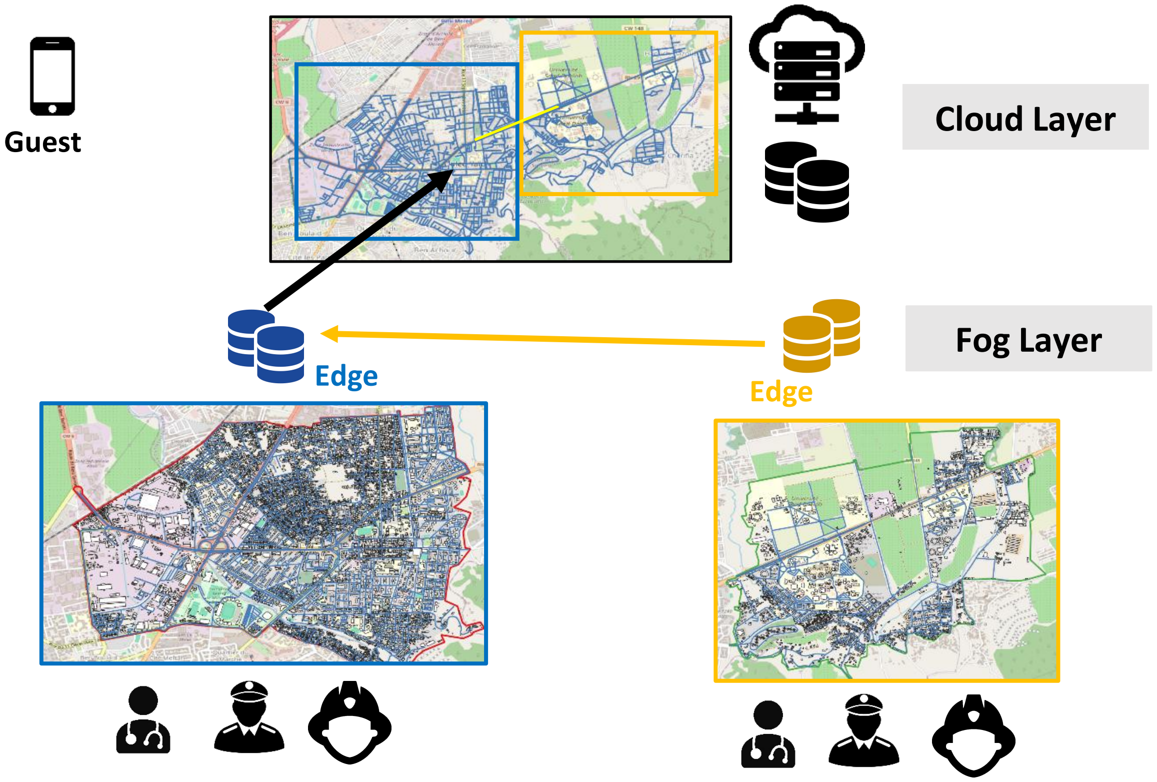


Figure 20: Architecture of the prototype

- **Requirements:**

The hardware and software requirements that were used for the prototype should be listed. The technical specification of the laptop that has been used to run the application are a processor AMD Ryzen 7 7840HS w/ Radeon 780M with Graphics 3.80 GHz and 15Go of RAM.

As for software specification, the project has been developed using Visual Studio Code with JavaScript language and the following libraries integrated into the project directories:

- Express.js: a web framework for Node.js.
- Pg: PostgreSQL client for Node.Js.
- Jquery: to facilitate event handling and HTML document manipulation.
- Body-parser: middleware module for Node.JS to handle POST and PUT requests, where data is typically sent in the request body.
- CORS: is an optional package to include as a support for server to handle CORS issues due to running it in a local environment.
- WS: Web Sockets to allow real-time communication between servers. Both socket.io and socket.io client dependencies have been included to enable bi-directional interactions.

Those are the main points that should be delineated for now, but with the challenges that will submerge, more software will come to answer and provide certain services that will allow to reach a more accurate implementation.

- **Back-end service design:** In this subsection, snippet of JavaScript Node.js are exhibited to present the way client side and server side along with other pieces of code all come together to provide a service and to be implemented, triggered by a simple

button to make it user-friendly. In an example, both of the buttons such as and “Share to my team” which saves every created feature, send it to the database, then add it back to the map so it gets broadcasted through opened interfaces.
It wasn’t evident to employ a service with this new architecture, because it’s not considered as best practice to load big segments of code all in one file. Therefore, it had to also be divided into small categories, all loaded in different JS files and excluding all necessary implementations to imply libraries or dependencies, it is apparent that there’s three main files to develop the set of actions the service requires:

- *Map.js:* this file contains the map initialization, drawing and styling features and where functions that adds and delete GeoJSON data should be integrated.
- *HandleButton.js*: This file where functions that are triggered by buttons, and data is fetched with a POST method to forward it to the server.
- *Server.js*: Responsible of establishing connection with the database, receiving fetched data and inserting it in the appropriate PostgreSQL table.

And here’s how the result came going from what each file handles (Figure 21)

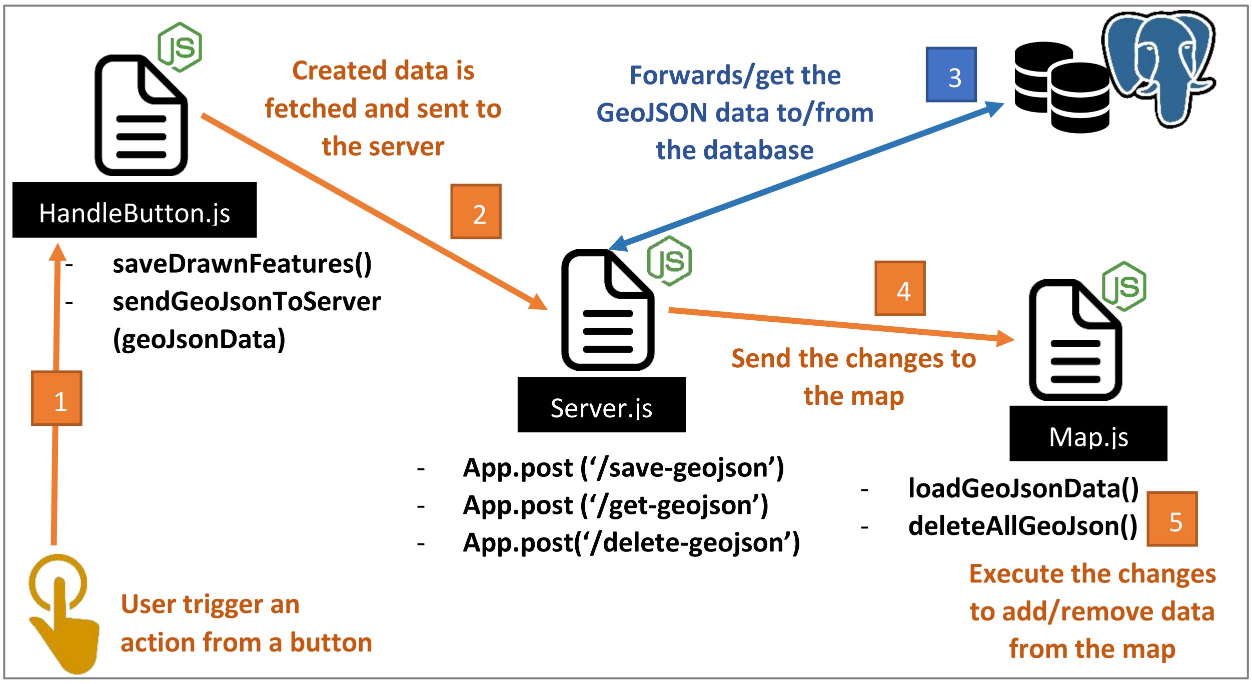


Figure 21: *Example on a back-end execution scenario triggered by the user to showcase server-side interactions with different client-side codes and PostgreSQL database to provide a service.*

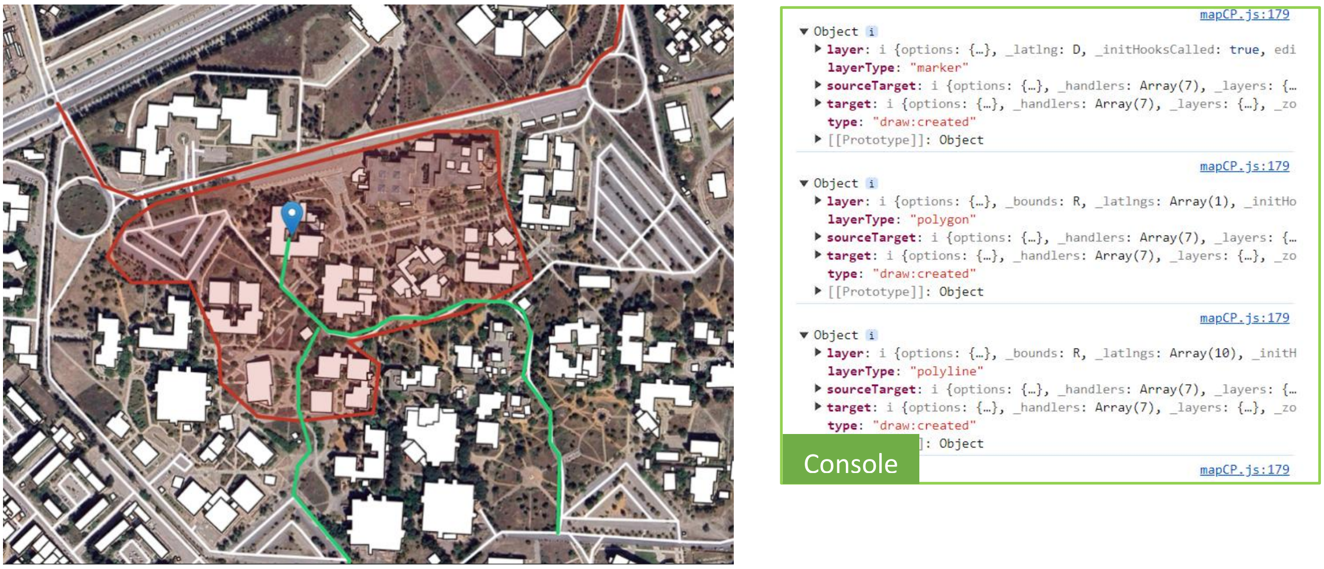
As an addition, to make the last step of the process more comprehensive and amplify how does the map update its collection data. The map.js file simply fetch from the post method executed by the server with the specification of the port that the server is currently running and listening to assure its functionality, since declaring only the post method isn’t enough and would likely generate a 405 HTTP error signaling its none existence.

Taking this simplified process, when the concept of socket gets added to the design, only the first part of the code gets replaced. The client request is generated from the button, the object is then gets sent to the server that assures the saving process. After the vectorial data has been saved, it will simplify the process of fetching it, return it as a result set that will be defined as a new overlay layer to add on top of the map to be displayed to all users.

III. Execution and performance

The key performance indicators to prove any system's efficiency are to emphasize the database, focusing on metrics such as transaction rates, tuple processing, and overall system throughput. By analyzing these metrics, valuable insights are gained into the database, its capacity to manage concurrent operations, and its responsiveness under varying loads.

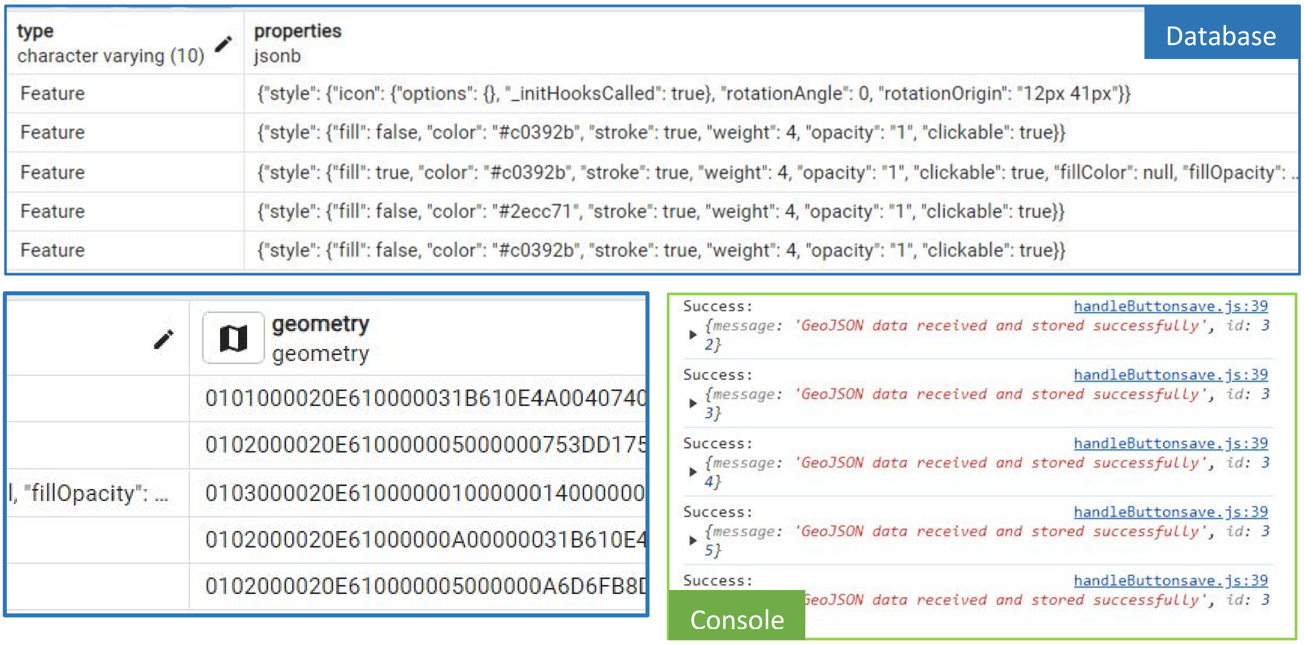
As discussed in previous chapters, it has been noted that each edge can have different types of users: civil protection, police officers, and urgent medical services end users. Each has a set of responsibilities, duties, and privileges that differentiate them from each other. However, for this execution, only the interactions of a civil protection agent were illustrated, considering the numerous extra features they authorize. In the first instance (Figure 22), after the participants have logged in, and in the case of a disaster, they start by generating crucial data that will soon be converted into a GeoJSON object, just as the console demonstrates.



```
                                                                mapCP.js:179
▼ Object i
  ▶ layer: i {options: {…}, _latlng: D, _initHooksCalled: true, edi
    layerType: "marker"
  ▶ sourceTarget: i {options: {…}, _handlers: Array(7), _layers: {…
  ▶ target: i {options: {…}, _handlers: Array(7), _layers: {…}, _zo
    type: "draw:created"
  ▶ [[Prototype]]: Object
                                                                mapCP.js:179
▼ Object i
  ▶ layer: i {options: {…}, _bounds: R, _latlngs: Array(1), _initHo
    layerType: "polygon"
  ▶ sourceTarget: i {options: {…}, _handlers: Array(7), _layers: {…
  ▶ target: i {options: {…}, _handlers: Array(7), _layers: {…}, _zo
    type: "draw:created"
  ▶ [[Prototype]]: Object
                                                                mapCP.js:179
▼ Object i
  ▶ layer: i {options: {…}, _bounds: R, _latlngs: Array(10), _initH
    layerType: "polyline"
  ▶ sourceTarget: i {options: {…}, _handlers: Array(7), _layers: {…
  ▶ target: i {options: {…}, _handlers: Array(7), _layers: {…}, _zo
    type: "draw:created"
                  ]: Object
                                                                mapCP.js:179
```

Figure 22: *Execution phase: Data creation by civil protection agent*

Once the entire map is set with the new layer, which can also be considered a feature, all elements are saved to the database successfully. This ensures they can be fetched again to display back on the map, not only locking them in the base map layer but also making them visible to all users interacting with the same server and assigned to the same Edge area (Figure 23).



type character varying (10)	properties jsonb
Feature	{"style": {"icon": {"options": {}, "_initHooksCalled": true}, "rotationAngle": 0, "rotationOrigin": "12px 41px"}}
Feature	{"style": {"fill": false, "color": "#c0392b", "stroke": true, "weight": 4, "opacity": "1", "clickable": true}}
Feature	{"style": {"fill": true, "color": "#c0392b", "stroke": true, "weight": 4, "opacity": "1", "clickable": true, "fillColor": null, "fillOpacity": …
Feature	{"style": {"fill": false, "color": "#2ecc71", "stroke": true, "weight": 4, "opacity": "1", "clickable": true}}
Feature	{"style": {"fill": false, "color": "#c0392b", "stroke": true, "weight": 4, "opacity": "1", "clickable": true}}

	geometry geometry
	0101000020E610000031B610E4A0040740
	0102000020E610000005000000753DD175
l, "fillOpacity": …	0103000020E61000000100000014000000
	0102000020E61000000A00000031B610E4
	0102000020E610000005000000A6D6FB8D

```
Success:                                            handleButtonsave.js:39
▸ {message: 'GeoJSON data received and stored successfully', id: 3
  2}
Success:                                            handleButtonsave.js:39
▸ {message: 'GeoJSON data received and stored successfully', id: 3
  3}
Success:                                            handleButtonsave.js:39
▸ {message: 'GeoJSON data received and stored successfully', id: 3
  4}
Success:                                            handleButtonsave.js:39
▸ {message: 'GeoJSON data received and stored successfully', id: 3
  5}
Success:                                            handleButtonsave.js:39
           eoJSON data received and stored successfully', id: 3
```

Figure 23: *Execution phase: Result of broadcasting to all users of the same Edge by saving the GeoJSON data to the database then adding it to the map.*

After the set of actions has been performed, it will be possible to examine the first case of recorded statistics in the dashboard that comes integrated into PostgreSQL by default. In addition to recording database transactions, the system's performance can also be monitored. By installing the extension "system_stats" in the database, system efficiency can be recorded by studying metrics such as memory usage, CPU utilization, and more. However, since all interfaces and end-users' instances are running locally on the same machine, this extension might not provide many insights. Therefore, the focus has been directed toward database activities.

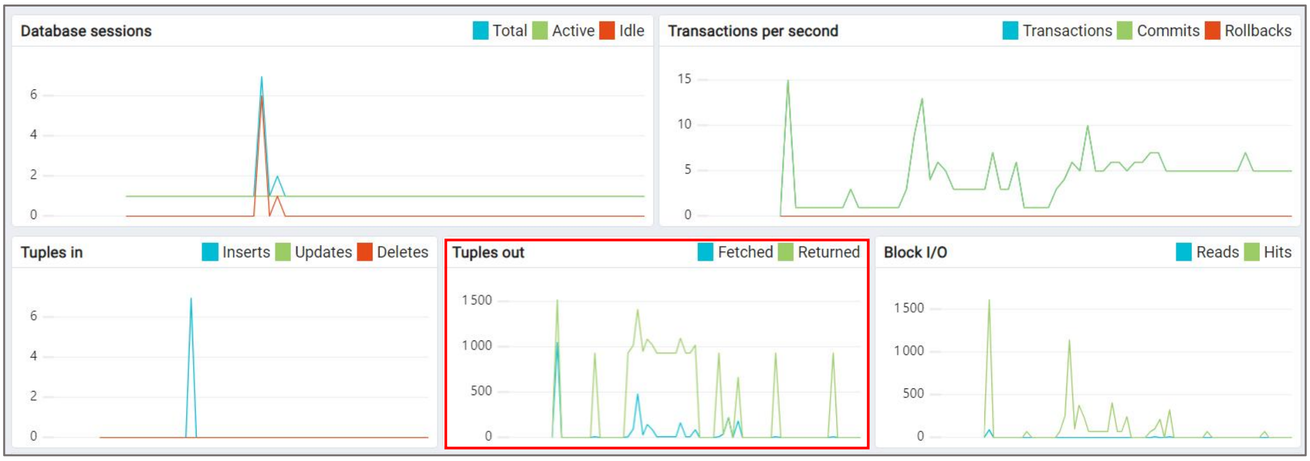


Figure 24: *Recorded statistics in the case of inserting new objects to the database.*

When inserting new rows into a table, the main metric to refer to is the number of tuples out. High peaks suggest that the system was actively modifying the database content, reflecting robust data handling capabilities. Additionally, to differentiate between tuples fetched and tuples returned, it is important to note that tuples fetched indicate the actual physical reads performed to satisfy the query executed by the server, while tuples returned show the number of rows that are returned to the client as the result set.

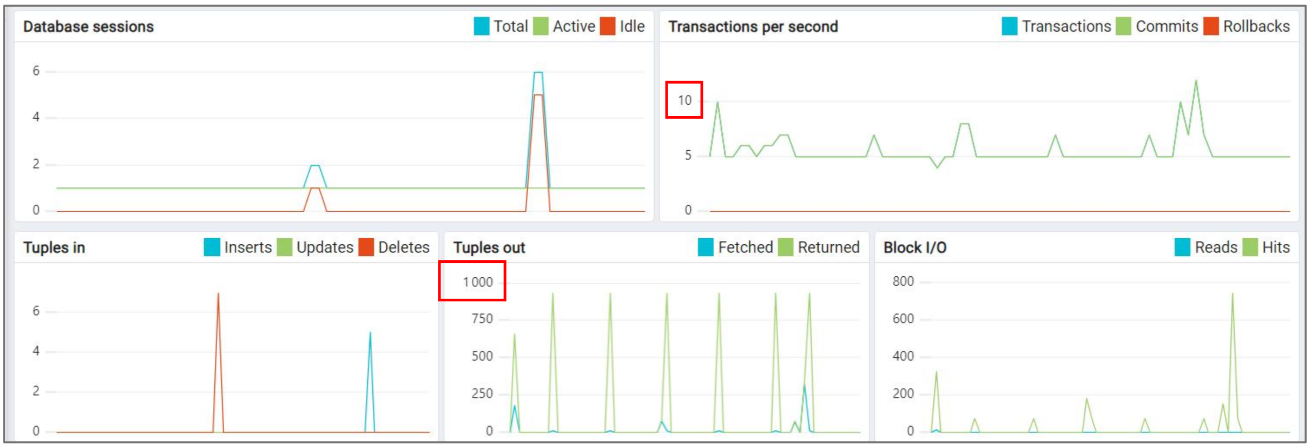


Figure 25: *Recorded statistics in the case of deleting recent added data collection from the database.*

Conversely, in (Figure 25), a notable decline is observed when attempting to delete rows from a table. Both transactions per second and tuples out recorded low values, indicating that the system requires minimal processing capacity to return the result set, in comparison to the first statistics noted in (Figure 24). It is evident that the system is capable of maintaining high performance levels even when confronted with the challenge of managing large segments of code and loading complex objects such as GeoJSON. This prototype is therefore capable of coordinating map layers and potential new layers created.

IV. Future directions

By embracing innovation and exploiting the potential of new technologies and strategies, disaster management can become even more effective, ultimately safeguarding the well-being and safety of entire nations. As threats from natural hazards intensify due to climate change, a proactive approach that capitalizes on every available advancement is crucial for building a more resilient future for all. Thenceforth, the proposed research ending up by the prototype can be considered only as foundation to develop a more significant DM system. The primary objective of this research was to explore the promising advantages offered by Geographic Information Systems, Virtual Reality, and Edge computing.

The interface of the proposed system can be enhanced by incorporating dual map rendering capabilities. The first type of map, which has already been implemented, is instrumental in covering the response and recovery phases of disaster management. This map provides valuable real-time data and situational awareness crucial for immediate decision-making and coordination during and after a disaster.

To further augment the system, a second type of map render can be integrated, offering a three-dimensional view achieved through the use of Rasters and elevation data. This advanced rendering can significantly improve the spatial analysis capabilities of the system. With the integration of remote sensing technologies, which are widely used in conjunction with GIS for disaster management, this 3D view can provide a more comprehensive understanding of the terrain and potential impact areas.

In succession, the second render that can be implied into the system is a three-dimensional view realized by using the concept of Rasters and Elevation data. With the aid of Remote sensing, one of the most significant technologies for enhancing GIS capabilities, can be leveraged to monitor environmental changes and detect alterations in earth levels. By fusing remote sensing data with GIS, the system can offer improved spatial analysis, feature extraction, and predictive modeling. This integration is vital for addressing all phases of disaster management, from mitigation and preparedness to response and recovery.

The proposed research and prototype development lay the groundwork for a more advanced disaster management system. By integrating technologies such as GIS, VR, and remote sensing, the system can provide comprehensive support across all phases of disaster management. The dual map rendering capabilities, combined with enhanced spatial analysis and predictive modeling, will ensure that disaster management efforts are more efficient, effective, and resilient. This proactive and technologically advanced approach is essential for safeguarding communities and building a resilient future in the face of increasing natural hazards.

V. Conclusion

This chapter has conducted a comprehensive analysis and has outlined potential directions for further development. It has not only demonstrated the system's achievements in creating a geographical information system and managing geospatial data, but also identified areas for improvement, particularly in opting for the Edge Computing concept in order to achieve more accurate and precise representation, and to enable web mapping. Practical solutions have been proposed to enhance the system's reliability, ensuring its smooth operation even under demanding conditions. One such solution is to split the code into partitions and include them where they won't require a redefinition to assure flawless execution.

One such direction is the evolution of the system to serve broader and more complex purposes. By integrating additional features, such as overlaying maps, the system can expand its scope and impact, addressing larger societal needs. Furthermore, it can have the possibility to disable layers, which would make it easier to contour low network capacities. This forward-looking approach underscores the importance of continued innovation and development. By staying at the forefront of technological advancements, we can ensure the system remains relevant and capable of tackling significant challenges such as implementing a three-dimensional render for disaster preparedness and mitigation.

In conclusion, the potential for integrating new ideas and concepts is considerable, to create a system that is both more extensive and efficient. This dedication to ongoing development will allow the system to serve as a versatile solution for significant societal issues, ultimately contributing to the betterment of society.

Conclusion

In this concluding chapter, the extensive research and development efforts presented throughout this thesis is synthesize. The objective is to provide a thorough summary of the realization of the proposed solution, highlight the significant contributions made, and draw general conclusions about the overall effectiveness and impact of our work.

- **Limitations**

The main barrier that was discovered is trying to design a physical network with numerous computers connected to a single edge as a starter, it was meant to emulate real-world user interactions. While virtual machines (VMs) proved to be inefficient due to the strain they put on actual hardware, especially when running resource-intensive applications such as our application that uses a mainly combination Geographical Information Systems with Virtual Reality. This method has shown to be ineffective for our goals.

To overcome this, Docker has been used during the conception of the solution, an open-source containerization allowing us to simulate different machines within a MACvlan network, created by assigning mac addresses to each container to replicate a physical environment. Unfortunately, this proposition also caused difficulties such as the existence of multiple system resources in each container and network instability. Furthermore, the high storage requirements that could not be satisfied and uneven network performance during demos had to discard this method as well.

Finally, the program was executed locally, with multiple user instances on a single edge, to showcase the potential of integrating GIS, Edge Computing, and VR. Despite the technological obstacles, this strategy effectively demonstrated the promise of combining these technologies while also emphasizing areas for future improvement.

- **Implementation and validation**

Beginning with the practical implementation and validation of the prototype, a Geographic Information System was designed. This GIS utilized a base map and an overlay map containing vector data. The vector data plays a crucial role in enhancing visualization and decision-making. For example, by adding an occupancy attribute to each building in the area during a disaster, it becomes evident which buildings require priority for rescue efforts, maximizing the number of lives saved in a short timeframe. Similarly, highlighting industrial areas with polygons allows for the identification of potential hazardous or noxious waste. Contact with fire in such areas could lead to dreadful consequences, necessitating immediate evacuation.

Beyond disaster recovery, this solution can be used daily. For example, police departments could utilize it as an interface for traffic management, specifically to announce temporary road closures due to construction. Consider the recent closure of "the coast" announced on June 24th for six months. An accurate announcement broadcasted to citizens in the area could showcase the deviation plan with simple clicks, highlighting the solution's potential.

Therefore, it becomes apparent that the proposed solution can be easily tailored to serve multiple purposes by adding extra fields, descriptions, and focusing on data collection efforts. Returning to map visualization, which Virtual Reality gets incorporated. While the prototype used high-resolution satellite imagery for stability testing, a final product would benefit from a more balanced resolution to avoid display issues and sluggish response times.

Furthermore, achieving a high level of smoothness and reduced delays can be achieved through Edge computing. This involves splitting the map and data sets and assigning them to local servers. This approach facilitates complex geospatial data analysis by leveraging a socket architecture for communication. Two types exist:

- Traditional Client-Server within the Edge: Clients send requests to the server, which responds with results.
- Bi-directional Socket Architecture: Enables data flow in both directions between edge servers, reducing time-consuming processes like requesting data from both sides simultaneously.

Here's the key difference from a normal client-server architecture: In this prototype, designing an interactive map with numerous features necessitates dividing the code itself to ensure functionality and efficient operation.

- **Contribution by contouring other systems and studies' weakness areas**

Following the realization details, the primary contributions of this solution must be outlined. The theme itself makes it apparent that the proposed solution is inherently "cooperative." By leveraging Edge Computing's advantages, the system only loads relevant segments of the database. This approach significantly outperforms fetching the entire dataset, especially when compared to traditional Cloud computing with a single central server. Additionally, the system exhibits "low-speed network tolerance" by allowing users to disable specific map layers when encountered with poor connectivity.

Finally, the solution aligns with the previously stated concept in implementation section of "GIS for disaster recovery." The success of this approach is evident when compared to similar systems like "Three-dimensional large-scaled GIS" that utilizes satellite imagery and elevation data. However, such systems struggle to achieve accuracy when rendering large areas, as they cannot load elements like bridges or rural roads.

This proposition is not the only one, many studies throughout the years have highlighted issues with rendering, realism, and mimicking real-world layouts in Geographic Information Systems. These studies consistently face the challenge that their approaches cannot encompass every aspect of a real map. In contrast, our approach can easily achieve a high degree of accuracy by integrating recent concepts.

- **Future path**

Wrapping up by discussing the future path and how it can improve the proposed solution, it is important not to forget that disaster management is divided into four phases: mitigation, prevention, response, and recovery. Since this research aimed to create a system for recovery, the next logical step would be to try to encompass the other stages and develop a complete disaster management system. This system would not only facilitate decision-making during a disaster but also help prevent and mitigate them.

Starting with mitigation, which involves studying past experiences to establish preventive laws or regulations, implementation can be relatively straightforward. One example is to save the vectorial data layer in a separate table named "archive" with a unique field (e.g., date or description) for each disaster management case other than deleting it as soon as the disaster has been put under control. This simplifies browsing, studying past decisions, and improving them for future situations.

Prevention has been addressed and implemented by many approaches before, by integrating Remote Sensing with Geographic Information Systems. From this research perspective, an incorporation could be achieved by creating two user-switchable interfaces. The first remains the existing flat surface satellite imagery and vectorial data used for response and recovery. The second interface would utilize Digital Elevation Models (DEMs), or more specifically Raster data, to create a gridded map with sensor-recorded statistical values for tracking vital signs of potential disasters. Existing implementations like thermal maps, which record temperature changes for wildfire detection. Additionally, incorporating altitude would provide a three-dimensional visualization of the map, allowing recording of any earth level alterations. It's important to consider the research implementation area size. Smaller areas simplify data recording, while larger areas will overburden recording equipment and complicate maintenance.

References:

(1) Khan, S.M.; Shafi, I.; Butt, W.H.; Diez, I.d.l.T.; Flores, M.A.L.; Galán, J.C.; Ashraf, I. A Systematic Review of Disaster Management Systems: Approaches, Challenges, and Future Directions. Land **(2023)**, 12, 1514. [DOI]

(2) Fei Wang *et al* **(2021)** *J. Phys. : Conf. Ser.* 2138 012027DOI 10.1088/1742-6596/2138/1/012027

(3) Hua Wang, Yue Wu, Xu Han, Mingliang Xu, Weizhe Chen: Automatic generation of Large-scale 3D Road Networks based on GIS Data (2021) [DOI]

(4) Kemper, H. and Kemper, G.: SENSOR FUSION, GIS AND AI TECHNOLOGIES FOR DISASTER MANAGEMENT, Int. Arch. Photogram. Remote Sens. Spatial Inf. Sci., XLIII-B3-**2020**, 1677–1683, (2020) [DOI]

(5) Cao, Y.; Xu, C.; Aziz, N.M.; Kamaruzzaman, S.N. BIM–GIS Integrated Utilization in Urban Disaster Management: The Contributions, Challenges, and Future Directions. Remote Sens. **(2023)**, 15, 1331. [DOI]

(6) YN Dwith Chenna: Edge AR/VR: exploring opportunities and confronting challenges. (**2023**) [REF]

(7) Abid, S.K.; Sulaiman, N.; Chan, S.W.; Nazir, U.; Abid, M.; Han, H.; Ariza-Montes, A.; Vega-Muñoz, A. Toward an Integrated Disaster Management Approach: How Artificial Intelligence Can Boost Disaster Management. Sustainability **(2021)**, 13, 12560. [DOI]

(8) Fazeldehkordi, E.; Grønli, T.-M. A Survey of Security Architectures for Edge Computing-Based IoT. IoT **(2022)**, 3, 332–365. [DOI]

(9) By WEISONG SHI, Fellow IEEE Guest Editor GEORGE PALLIS Guest Editor ZHIWEI XU Guest Editor: Edge computing: scanning the issue 10.1109/JPROC.**2019**.2928287

(10) Prieto, J.; Durán Barroso, R.J. Emerging Technologies in Edge Computing and Networking. Sensors **(2024)**, 24, 1271. [DOI]

(11) Liu J P Zhang J S, Wang H, Sun X L And Zhu L L **(2018)** Research on 3D GIS Platform Construction of Expressway Based on 3D Data Model 2018 International Conference on Smart Grid and Electrical Automation (ICSGEA) pp 341-345

(12) Zhu Q, Li X, Cao D **(2013)** GIS-Based 3D Modeling and Visualization in Luohe Basin 2013 International Conference on Information Technology and Applications pp 328-332

(13) Alena Yablokova, Dmitry Kovalev, Igor Kovalev and Valeria Podoplelova: Standards and technologies of Web development when designing the GUI of web-based geographic information systems of the agro-industrial complex (**2024**) [DOI]

(14) Barry Evans, Clive E Sabel: Open-source web-based geographical information system for health exposure assessment (**2012**)

(15) Atif Naseer, Hossam Aldoobi, Basem Y.Alkazemi : A service-oriented architecture for GIS applications (**2015**) [DOI]

(16) Fritz Kleinschroth a, *, Kawawa Banda b and al: Drone imagery to create a common understanding of landscapes (**2022**) [DOI]

(17) Sun, W.; Bocchini, P.; Davison, B.D. Applications of artificial intelligence for disaster management. Nat. Hazards (**2020)**, 103, 2631–2689. [REF]

(18) Docteur de faculté des sciences politiques et relations internationales Djouhri Smail Cours : PLANS D'URGENCE ET GESTION DES CRISES (**2021**) [REF]

(19) FICHE DE PLAN ORSEC COMMUNE (11 FICHES) [REF]

(20) Mme ASSAM Nadia – Sciences économique, commerciales et sciences de gestion : LA DEFENCE CIVILE EN ALGERIE de la revue algérienne des sciences Juridiques, Economiques et politiques. [REF]

(21) Journal officiel de la république algérienne démocratique et populaire N°10 sur le plan ORSEC p6-11 [REF]

(22) Liang Tian, Xiaorou Zhong : A Case Study of Edge Computing Implementations: Multi-access Edge Computing, Fog Computing and Cloudlet (**2023**) [DOI]

(23) Paweł Grabarczyk: What is real in Virtual Reality? (**2024**) [DOI]

(24) Fethi Cekinmez : Radio Access Network [REF]

(25) Jaime Vergara, Juan Felipe Botero, Luis A. Fletscher : A Comprehensive Survey on Resource Allocation Strategies in Fog/Cloud Environments (**2023**) [REF]

(26) Open Buildings : A dataset of building footprints to support social good applications. [REF]

(27) QGIS Training Manual : Working with Raster Data [REF]

(28) OpenTopography : High-Resolution Topography Data and Tools [REF]

(29) Leaflet Open-source Javascript Library for interactive mapping [REF]

(30) Leaflet provider : a Github repository to list all open-source tile layer maps for web mapping [REF]