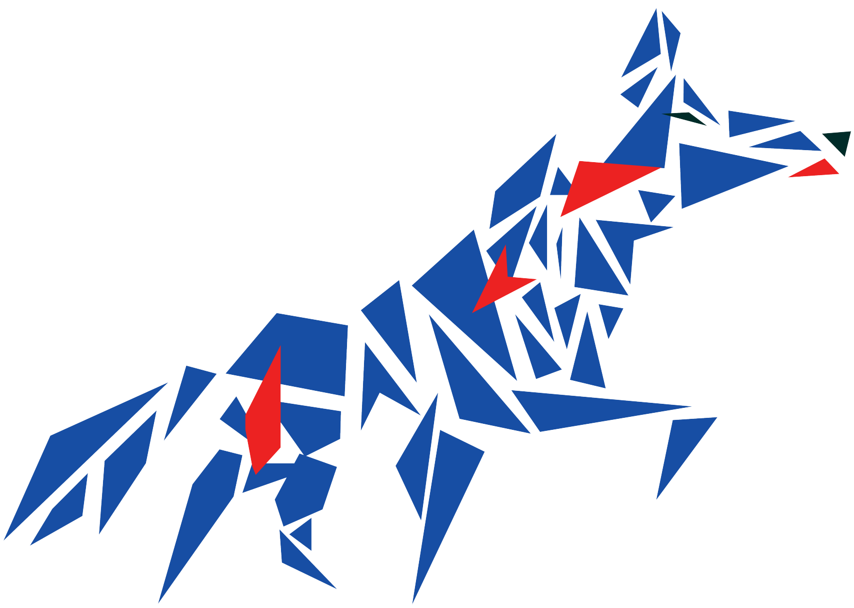
Innovative Smart System

Portefolio

Clément Gauché



Institut National des Sciences Appliquées de Toulouse

February 1, 2025

Abstract

This portfolio presents my academic and professional journey as part of my specialization in Innovative Smart Systems (ISS) at the National Institute of Applied Sciences of Toulouse (INSA Toulouse). It highlights the skills I have acquired through my coursework, innovative projects, and work-study experiences. The topics covered span a broad range of technologies, including embedded systems, wireless sensor networks, service-oriented architecture, artificial intelligence applied to IoT, energy management for autonomous systems, and cybersecurity of connected devices. Through this portfolio, I aim to demonstrate how these experiences have shaped my technical expertise and how they will contribute to my professional career.

Contents

I. Introduction **3**

II. Courses description **5**

1. Layout of the year's courses . . . 5
2. Apprenticeship at Vitesco Technologies . . . 7
   - 2.1 Context . . . 7
   - 2.2 Technical Summary . . . 7
   - 2.3 Skills Acquired . . . 7
   - 2.4 Analysis and remarks . . . 8
3. Innovative Project - Wispers . . . 10
   - 3.1 Context . . . 10
   - 3.2 Technical Summary . . . 10
   - 3.3 Skills acquired . . . 11
   - 3.4 Analysis and remarks . . . 11
4. Cloud and Edge Computing . . . 13
   - 4.1 Context . . . 13
   - 4.2 Technical Summary . . . 13
   - 4.3 Practical work . . . 13
   - 4.4 Skills acquired . . . 14
   - 4.5 Analysis and remarks . . . 14
   - 4.6 Reflections . . . 15
5. 5G technologies . . . 16
   - 5.1 Context . . . 16
   - 5.2 Technical and practical work . . . 16
   - 5.3 Skills acquired . . . 16
   - 5.4 Analysis and remarks . . . 16
6. Service Oriented Architecture - Software Engineering . . . 18
   - 6.1 Context . . . 18
   - 6.2 Technical Summary . . . 18
   - 6.3 Practical work . . . 19
   - 6.4 Skills gained . . . 20
   - 6.5 Analysis and remarks . . . 21
7. Wireless Sensor Networks . . . 23
   - 7.1 Context . . . 23
   - 7.2 Technical Summary . . . 23
   - 7.3 Practical Work . . . 23
   - 7.4 Skills acquired . . . 24
   - 7.5 Analysis and remarks . . . 25
8. Middleware for IoT . . . 26
   - 8.1 Context . . . 26
   - 8.2 Technical Summary . . . 26
   - 8.3 Practical Work . . . 26
   - 8.4 Skills Acquired . . . 27
   - 8.5 Analysis and Remarks . . . 28
9. Security for Connected Objects . . . 29

9.1 Context . . . 29
9.2 Technical Summary . . . 29
9.3 Practical Work . . . 30
9.4 Skills Acquired . . . 32
9.5 Analysis and Remarks . . . 33
10. Energy for connected objects . . . 34
10.1 Context . . . 34
10.2 Technical Summary . . . 34
10.3 Practical work . . . 34
10.4 Skills acquired . . . 34
10.5 Analysis and remarks . . . 35
10.6 Reflections . . . 35
11. LAB at AIME . . . 36
11.1 Context . . . 36
11.2 Technical Summary . . . 36
11.3 Practical Work . . . 36
11.4 Skills Acquired . . . 36
11.5 Analysis and remarks . . . 37
12. Microcontrollers, Open-Source Hardware, and Sensor Introduction . . . 38
12.1 Context . . . 38
12.2 Technical Summary . . . 38
12.3 Skills Acquired . . . 39
12.4 Analysis and remarks . . . 40
13. Communication protocols for LP-WPAN . . . 41
13.1 Context . . . 41
13.2 Technical Summary . . . 41
13.3 Practical Work . . . 42
13.4 Skills acquired . . . 42
13.5 Analysis and remarks . . . 42
14. Embedded AI for IoT . . . 44
14.1 Context . . . 44
14.2 Technical Summary . . . 44
14.3 Practical Work . . . 44
14.4 Skills Acquired . . . 45
14.5 Analysis and Remarks . . . 45
15. Master REOC . . . 46
15.1 Context . . . 46
15.2 Technical summary . . . 46
15.3 Skills acquired . . . 46
15.4 Analysis and remarks . . . 48

III.Conclusion **49**

Chapter I.

Introduction

Presentation

My name is Clément Gauché, and I am a fifth-year student at the Institut National des Sciences Appliquées de Toulouse, specializing in Innovative Smart Systems.
In this portfolio, I present the skills and experiences I have acquired in this field. Its purpose is to illustrate how my professional and academic experiences have enabled me to develop both technical expertise and cross-disciplinary skills. This portfolio highlights the courses I have taken, the innovative projects I have completed, and the work-study program I undertook in a company. For each subject, I aim to express what I have gained from it and how it fits into my academic and professional journey.

Portfolio Organization

This portfolio is organized to provide a clear and accessible overview of my academic and professional projects.
Each section corresponds to a specific subject and is divided into several key parts:

- Context: Introduction to the subject or project, explaining the objectives and general framework.
- Technical Summary: A summary of the theoretical and technical concepts covered.
- Practical Work: A description of the practical activities and projects undertaken.
- Skills Acquired: A presentation of the skills developed, illustrated with dedicated tables.
- Analysis and Remarks: Reflections and feedback on the course.

The skills acquired are categorized into two types of tables. The first type, containing the columns "Required" and "Achieved," is a self-assessment skills table provided by the instructors.

Skills	**Required**	**Achieved**
Self-evaluation with portfolio		
Reflect upon my training process and methods	3	4
Be able to put forward my training experiences, whether they are explicit or implicit	3	4
Be self-sufficient and responsible towards my education	4	4

Table I..1: Skill matrix for portfolio management and self-evaluation

The second type, containing the column "Level of Mastery," is a complementary self-assessment skills table that reflects the skills I consider to have acquired.

Chapter II.

Courses description

1. Layout of the year's courses

Date	Duration	Context	Content
from 15/06 to 22/09/2024	576 hours	Internship at IASI in Romania	Discovery and activation of the Hardware Security Module (HSM)
from 30/09/2024 to 29/01/2025	68.5 hours	Innovative Project	The guideline industrial project of our year
from 30/09 to 15/10/2024	21.5 hours	Cloud and Edge computing	Learn and practice about virtualization techniques such as VMs and containers
from 30/09 to 03/12/2024	7.5 hours	5G Technologies	Learn about 5G and more generally topics related to cellular networks
from 04/10/2024 to 14/01/2025	33.5 hours	Service Architecture	Learn about legacy and modern architectures service oriented for software engineering
from 07/10 to 14/12/2024	18.25 hours	Wireless Sensor Networks	Learn about wireless sensor networks technologies
from 15/10 to 09/12/2024	14.75 hours	Middleware for IoT	Discover communication protocols for IoT
from 15/10 to 24/01/2025	34.5 hours	Security for connected objects	Discussion about the need and how to secure protocols for IoT
from 07/10 to 17/10/2024	10.5 hours	Energy for Connected Objects	Introduction to the different methods to power Energy for Connected Objects

Date	Duration	Context	Content
from 04/11 to 08/11/2023	13.75 hours	Lab at AIME	Discover the creation of a sensor
from 12/11 to 18/12/2024	39 hours	Microcontrollers Open-Source Hardware and Sensor Introduction	Introduction to microcontrollers programming and implementation of our sensor in a complex circuit
from 04/12/2024 to 07/01/2025	7.5 hours	Communication protocols for LP-WPAN	Definition of a TCP/IP protocol stack for an LP-WPAN network
from 04/12/2024 to 22/01/2025	15.75 hours	Embedded IA for IoT	Introduction to the concepts of AI applied to an IoT context
from 08/01 to 24/01/2024	10.5 hours	Emerging network (SDN, NGN)	Discovery of emerging network paradigms
from 05/10 to 31/01/2024	6.75 hours	Portfolio	Writing of the document you are actually reading

Figure II..1: Table with information about all my projects

2. Apprenticeship at Vitesco Technologies

2.1 Context

During my second year of apprenticeship at Vitesco Technologies, my projects focused on embedded software development, specifically on the design of generic drivers for microcontrollers. For instance, this year I worked on developing a PWM driver for a new microcontroller in Infineon's AURIX family, the TC4. I also carried out integration tests for a project I had worked on the previous year: the Wake-Up Controller. As part of my degree requirements, I had to complete an internship in Iasi, Romania, where I worked on the initial activation of the Hardware Security Manager (HSM) for a Renesas microcontroller.

2.2 Technical Summary

Developing the PWM driver for Infineon's TC4 microcontroller was the only new task I took on within the Complex Device Driver (CDD) team this year. I had to design a generic driver that would enable simple and efficient control of PWM signals. I also performed unit and integration tests to validate the proper functioning of the *Wake-Up Controller*, a project I had contributed to the previous year. The main function of this module is to wake the main microcontroller from a deep sleep state or from a complete shutdown. To achieve this, I developed and applied test scenarios to validate wake-up triggers based on analog signals.

Finally, I completed a three-month internship in Iasi, Romania, during which I worked on activating the Hardware Security Manager (HSM) for a Renesas microcontroller, the RH850. I needed to understand how this module operates and carry out its initial activation, while ensuring it could still be unlocked. Indeed, in theory, once the HSM is activated, reprogramming the microcontroller is no longer possible.

2.3 Skills Acquired

Throughout my apprenticeship, I developed a variety of technical and professional skills:

Skill	Description	Level of Mastery
Embedded Systems Development	Proficient in designing and implementing drivers for embedded systems (e.g., PWM driver development).	Advanced
Microcontroller Programming	Knowledge of microcontroller architectures (e.g., Infineon TC4, Renesas RH850).	Advanced
Software Validation	Experience in unit, integration, and black-box testing of embedded modules.	Advanced
Hardware Security Implementation	Hands-on experience activating and configuring Hardware Security Modules (HSM) for secure communication.	Intermediate
Low-Level Programming	Writing and debugging low-level code, including linker configuration and interrupt handling.	Advanced
Software Engineering Principles	Applied modular and scalable development practices, including BuildUnit management.	Advanced
Cross-Functional Communication	Experience conveying technical details to multidisciplinary teams (e.g., firmware, hardware, and testing groups).	Advanced
Documentation and Reporting	Created technical documentation and test reports for internal validation and project deliverables.	Advanced
International Work Experience	Gained cultural and professional exposure by working with multinational teams.	Intermediate

Table II..1: Skills acquired since the start of my apprenticeship at Vitesco Technologies

2.4 Analysis and remarks

Choosing an apprenticeship in my third year was a pivotal step in my journey. This experience allowed me to gain firsthand insight into the realities of industry and to better appreciate the importance of the courses taught at INSA. I was able to apply the knowledge acquired at school while developing new technical and professional skills. The projects I worked on covered a wide range of areas, from microcontroller driver development to activating a hardware security module. Furthermore, my stay in Romania, working with the Iasi teams, provided me with a rich cultural and professional immersion and allowed me to collaborate with international experts.

I found the Hardware Security Manager (HSM) activation project particularly interesting because it challenged me with secure system design and showed how the security principles studied in class are put into practice. I also learned to communicate more effectively and coordinate with geographically dispersed teams, which strengthened my project management and international collaboration skills. Getting up to speed with a new microcontroller such as the Renesas RH850 was a steep learning curve that required perseverance and teamwork to overcome.

Finally, balancing academic and professional obligations enhanced my time management skills and helped me to set clear priorities. This is why I consider my apprenticeship at Vitesco Technologies a major turning point in my engineering career: it allowed me to deepen my technical expertise, broaden my perspective on international teamwork, and confirm my interest in cybersecurity and security.

3. Innovative Project - Wispers

3.1 Context

The WISPERS (Wireless System for Intracranial Pressure Monitoring based on 3D-printed Piezo-capacitive Sensor) project is a research initiative developed in collaboration with LAAS-CNRS MEMS and MINC teams and INSA Toulouse. The project aims to improve intracranial pressure (ICP) monitoring through a wireless, battery-free system capable of continuous and precise data collection.

Current ICP monitoring techniques are invasive, costly, and pose infection risks, making it imperative to develop a safer and more efficient solution. WISPERS addresses this challenge by implementing a non-invasive skin patch that collects the intracranial sensor data and transmits it to a central hub via a secure, low-power, and robust wireless communication system.

Our team was responsible for the communication architecture, including:

- Implementing the RuBee protocol for secure data transmission.
- Developing the modulation and demodulation systems for signal transmission.
- Designing and testing the state machines to handle data exchange between the patch and the hub.
- Ensuring real-time visualization of the transmitted data through a web interface.

3.2 Technical Summary

To achieve wireless, battery-free data transmission, we adopted the RuBee communication protocol, which is known for its low power consumption, resistance to electromagnetic interference, and high security. Unlike Bluetooth or Wi-Fi, RuBee operates at low frequencies (131 kHz), making it ideal for medical applications where traditional RF protocols struggle due to metallic environments or biological tissues.
Our contributions focused on:

- Protocol Data Unit (PDU) construction and decoding in VHDL for the patch.
- State machine design for managing data exchange.
- Signal modulation and demodulation using Amplitude Shift Keying (ASK) and Bi-Phase Mark Coding (BMC) for enhanced robustness.
- Development of a digital filter to extract meaningful data from the received signals.
- Hardware design, including PCB prototyping and unitary testing of key components such as DAC (R2R ladder), inductance, and active filtering circuits.

Each software and hardware component was independently tested and validated, ensuring the correct functionality of each module. However, due to time constraints, we were unable to test the fully assembled system. The physical modulation part was successfully tested at the LAAS laboratory, confirming its feasibility for signal transmission.

3.3 Skills acquired

Skills	Required	Achieved
Manage an innovative project		
Solve a problem in a creative way	4	4
Develop the first stage of innovation	4	4
Understand production, validation, distribution, acceptability, and aftermath of innovation	4	4
Structure and lead an innovative project	4	4

Table II..2: Skill matrix for innovative project management and self-evaluation

Skill	Description	Level of Mastery
Innovative Project Management	Managing a long-term, interdisciplinary project in a medical context.	Advanced
Low-Power Wireless Communication	Implementing and optimizing the RuBee protocol for data transfer.	Advanced
Digital Signal Processing	Modulating, demodulating, and filtering signals for robust data transmission.	Intermediate
Hardware Design & Testing	Designing PCB components and validating them through unitary tests.	Intermediate
Embedded Systems (VHDL & C++)	Implementing PDU encoding/decoding and protocol logic in hardware.	Advanced
Medical IoT Development	Adapting wireless sensor technologies for biomedical applications.	Intermediate

Table II..3: Skill matrix for the WISPERS project

3.4 Analysis and remarks

This innovative project was one of the most challenging and rewarding experiences of my academic journey. It allowed me to apply advanced embedded system concepts to a real-world medical problem, requiring a high level of technical expertise, teamwork, and problem-solving.
One of the main difficulties we faced was the time constraint, which prevented us from fully assembling and testing the complete system. However, we successfully validated each module independently, ensuring that the communication protocol, modulation, and data handling worked correctly.

A key success was the physical modulation tests conducted at the LAAS laboratory, which confirmed the feasibility of using Amplitude Shift Keying (ASK) and Bi-Phase Mark Coding (BMC) with RuBee. These results were particularly satisfying, as they demonstrated the robustness of our signal transmission approach in a real medical environment.

Another valuable takeaway was the interdisciplinary collaboration required for this project. We had to coordinate between hardware, software, and communication systems, which reinforced my ability to work in a multidisciplinary team. Additionally, adapting low-power wireless communication to a biomedical application highlighted the importance of energy efficiency and secure data transmission in healthcare technologies.

If we had more time, we would have liked to integrate all hardware and software components into a fully functional prototype, and perform a complete end-to-end testing of the system, from data collection to web interface visualization.
We would have also liked to optimize energy consumption for longer operation without external power sources. And despite these limitations, this project has strengthened my interest in IoT and embedded systems. It has also provided valuable experience in protocol implementation, hardware testing, and digital signal processing, which will be invaluable for my future career in embedded systems and IoT security.

4. Cloud and Edge Computing

4.1 Context

Among the first courses I came across during my final year at 5ISS was Cloud and Edge Computing. This course introduced me to core concepts of both distributed and pervasive computing together with the analysis of a basic architecture that included techniques of Service Oriented Computing (SOC), virtualization techniques, and cloud technologies as enablers of utility and pervasive computing. By nature, the course had both a theoretical and hands-on combination that gave me an insight into how such paradigms shape the current IT systems and their applications.

4.2 Technical Summary

Virtualization has been amply covered in class. To be more specific, we discovered the different types of it: hypervisors, Type 1 and 2, paravirtualization, and containerization. We have also considered Virtual Machine Templates, which strongly ease virtual environments management and deployment. A specific highlight was the comparison between service orchestration and service choreography, where we analyzed the strengths and limitations of each. While orchestration does indeed ease updates by adopting a centralized approach, it tends to make the system slower. On the other hand, choreography is decentralized, enhancing autonomy but with a very painstaking design process. The choice between them is necessarily dependent on the specific contexts of development and economics in which they are applied.

The course also emphasized cloud computing by exploring definitions, including the widely recognized NIST framework, while identifying its core characteristics. We also looked at on-demand services, scalability, pay-per-use models and the delivery of services through models such as Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). Additionally, we examined edge computing and its integral role in IoT ecosystems, highlighting its ability to reduce latency and uphold data sovereignty. However, this approach also comes with challenges, such as connectivity limitations and the constrained processing power of edge devices. And finally, we addressed fog computing as an intermediary layer that bridges cloud services with data sources, reducing the distance data must travel and enabling real-time interaction.
The professor defined this concept as a process of “cloudification of the IoT world”.

4.3 Practical work

The practical work for this course was divided into two major sessions, each designed to reinforce the theoretical aspects covered in class. In the first session, we explored cloud hypervisors by engaging with tools such as VirtualBox and the OpenStack API. This practical session allowed me to experiment with various virtualization techniques and understand their implementation in controlled environments.

The second session was on service orchestration in hybrid cloud and edge environments. In that exercise, I learned the deployment and management of a distributed system operating across these two architectures, thus providing insights into their complexities and potential.

These practical sessions were crucial for bridging the gaps in theory into applications. They provided me the chance not only to comprehend the conceptual aspects but also to apply the same in realistic scenarios. This solidifies my comprehension and confidence in handling Hybrid Architectures.

You can find the reports of these labs in the appendix (Reports folder).

4.4 Skills acquired

Skills	Required	Achieved
Cloud and Autonomic Computing		
Understand the concept of cloud computing	3	3
Use a IaaS-type cloud service	3	3
Deploy and adapt a cloud-based platform for IoT	3	3

Table II..4: Skill matrix for Cloud and Autonomic Computing

Competence	Description	Level of Mastery
Designing Cloud and Edge Architectures	Understanding of distributed and cloud architectures, with a focus on performance, scalability, and reliability in hybrid environments.	Intermediate
Virtualization Tools and Environments	Creation and management of VMs and containers using platforms such as VirtualBox and the OpenStack API.	Advanced
Container Orchestration	Deploying services on container orchestration platforms like Kubernetes, and ensuring efficient scaling and fault tolerance.	Intermediate
Resilient Systems for IoT	Ability to design systems that maintain low latency, high availability, and robust security, tailored for IoT and distributed scenarios.	Intermediate
Service Continuity at the Edge	Adept at managing edge node variability, handling real-time disruptions, and ensuring uninterrupted service delivery.	Intermediate

Table II..5: Skills Acquired in Cloud and Edge Computing

4.5 Analysis and remarks

What I liked about this course was the structuring of the practical sessions, which combined theory and practice quite well. Each session began with a theoretical introduction to reinforce the central concepts of the course. We then had to go through an implementation phase, putting these ideas into practice with tools such as VirtualBox and OpenStack APIs. In this way, my understanding of the course was reinforced and I was able to acquire “concrete” practical skills. I found this format particularly rewarding because it reflects real-world workflows, where a thorough understanding of concepts must complement practical expertise.

4.6 Reflections

This course gave me a better understanding of the interaction between cloud and edge computing systems and how they enable innovative, scalable, and efficient solutions. The smooth integration of theoretical and practical aspects has prepared me for the challenges of analysing the design and the management of distributed systems. As I plan to pursue a specialized master’s degree in cybersecurity, I recognize the value of understanding virtualization, container orchestration, and hybrid architectures in designing robust defenses against emerging threats.

Through the labs of this course, I can say that I developed not only my technical skills but also my critical thinking and problem-solving skills, both of which are I think important for my future career in cybersecurity.

5. 5G technologies

5.1 Context

The 5G technologies module presented by Professor Etienne Sicard, focused on the rapid evolution of cellular networks, emphasizing the transformative nature of 5G. A unique aspect of the learning approach was the use of reverse pedagogy, where students presented the majority of the content. The central goal was to explore cutting-edge topics in mobile communications, ranging from modulation techniques to the societal impacts of 5G and beyond.

5G represents a paradigm shift in mobile networks, introducing Software Defined Radio (SDR) and microservices, which revolutionized network architecture by increasing flexibility and efficiency. Through this course, we students, were encouraged to examine the current state of cellular technology while considering its implications for the future.

5.2 Technical and practical work

During this course we could discuss about many varied subject. I collaborated with Noel Jumin on a presentation titled "Samsung's Vision for 6G," analyzing how Samsung's ambitions for 6G diverged from its competitors. We highlighted Samsung's focus on sub-THz frequencies, AI integration, and next-generation services such as holographic communications. This required an in-depth understanding of how 5G technologies paved the way for these future advancements, including the challenges posed by scaling up.

Other students' presentations covered topics such as 5G modulation techniques, vehicular networks, and the environmental and societal implications of 5G and 6G. These presentations provided a comprehensive understanding of both the technical and non-technical aspects of mobile communications.

5.3 Skills acquired

Competence	**Description**	**Level of Mastery**
Mobile Communications Development	Understanding the major development phases for mobile communications and development of the associated technology.	Advanced
Impact of New Mobile Technology	Understanding the impact of new mobile technology.	Intermediate

Table II..6: Competences Gained During the 5G Technologies Module

5.4 Analysis and remarks

This course was highly engaging as it dealt with contemporary and practical topics in cellular network technology. The discussion of 5G as a foundational step towards 6G was particularly fascinating, showing how each generation builds upon the previous one. The reverse pedagogy approach fostered a dynamic and interactive learning environment.

However, I found reverse pedagogy less effective overall. While preparing and delivering our own presentations allowed us to master our chosen topic, it made it harder to fully comprehend the topics presented by others. During the presentations, my focus was often divided between understanding the other groups' content and refining my own work.

I believe that a better balance between lectures by the teacher and student presentations would enhance the learning experience. Teacher-led lectures could provide a deeper dive into each topic, ensuring a more comprehensive understanding, while our presentations could supplement this with detailed case studies or specific insights. As it stands, I feel I gained high-level knowledge of the subjects presented by others, but lacked the depth that direct teaching could have provided.

6. Service Oriented Architecture - Software Engineering

6.1 Context

The Service-Oriented Architecture (SOA) course was taught by Ms. Nawal Guermouche in the form of a MOOC. It introduces software engineering architectures and traces the evolution of software systems, from monolithic applications to distributed and modular architectures such as SOA, RESTful services and microservices. The course highlights the importance of this type of architecture in the creation of today's scalable, reusable and flexible software systems.

Thanks to this course, I discovered how the SOAP and WSDL protocols serve as the basis for SOA, enabling interoperability and modularity. I also learned about RESTful services, which offer a simpler, lighter approach to web services by relying on HTTP methods for communication. And finally, we discovered the principles of microservices, where the decomposition of large applications into smaller, independent services improves scalability, fault isolation and deployment flexibility.

As a student with a background in automation and electronics, this course was a great discovery for me and gave me a new perspective on software system design.

6.2 Technical Summary

The course looked at the principles and implementation of three key software architectures: SOA, REST and microservices. Service-oriented architecture (SOA) was presented as a modular design paradigm that encapsulates functionality in independent services, enabling loose coupling and reuse. Key technologies, such as SOAP (Simple Object Access Protocol) and WSDL (Web Services Description Language), were explored to understand how services are defined, discovered and invoked, thus offering seamless interoperability between platforms.

REST (Representational State Transfer) was presented as a lightweight alternative to SOAP, emphasizing stateless communication and the use of standard HTTP methods like GET, POST, and DELETE for interacting with resources. The simplicity and scalability of REST make it a popular choice for modern web services. In contrast, SOAP's complexity is offset by its robustness and suitability for enterprise-level systems.

Building on these foundations, the course introduced microservices as an extension of SOA, focusing on decomposing monolithic applications into smaller, independent services aligned with business functionalities. Each microservice operates autonomously, often managing its own database and exposing APIs, typically using REST. This approach facilitates scalability, continuous deployment, and fault isolation, making it ideal for large-scale, dynamic systems.

To complement these architectural concepts, the course provided practical insights into tools and technologies such as Spring Boot for building microservices, Postman for API testing, and Microsoft Azure for deploying distributed systems. These tools demonstrated how theoretical principles are applied in real-world scenarios to design and manage modern software systems effectively.

6.3 Practical work

The practical component of the course was structured into two stages: guided tutorials and independent projects. This approach allowed us to progressively build on theoretical concepts, starting with foundational exercises and advancing to complex implementations.

Guided Tutorials

The initial phase involved a series of tutorials, each focusing on a specific aspect of SOA, RESTful services, and microservices:

- SOAP: Introduction to SOAP web services, WSDL, and SOAP clients.
- REST: Setting up RESTful services, creating REST clients and handling data formats (e.g., JSON, XML)
- Microservices: Using Spring Boot and Spring Cloud to create services, manage service discovery, load balancing, configuration management, and client integration with configuration services.

These tutorials provided hands-on experience with tools and frameworks, helping us solidify our understanding of architectural principles.

Projects: Volunteering Application and Smart Building Simulation

After completing the tutorials, we applied our skills to two independent projects that simulated real-world scenarios:

The goal of the first project was to design RESTful microservices for a system where users could post requests for volunteers, volunteers could respond, and users could leave feedback. Using Java, we implemented multiple services, each responsible for specific tasks like request management, volunteer interactions, and feedback processing. This project emphasized the importance of well-structured service communication and the use of REST APIs for seamless integration. Smart Building Simulation:

In the second project, we created a distributed application that interfaced with a simulated sensor network for a smart building. This scenario included various rooms equipped with sensors (e.g., presence detectors) and actuators (e.g., doors, windows, alarms, lights). We developed the following microservices using Java:

- A sensor service to manage sensor data and events.
- An actuator service to control actuators based on sensor inputs.
- A room configuration service to handle building setups.
- A user service to determine the location of individuals within the building.
- A time service to synchronize operations across the system.

Additionally, we built a front-end interface using HTML, CSS, and JavaScript to visualize and interact with the smart building. A Python script simulated sensor behavior according to a predefined scenario, enhancing the realism of the simulation.

These projects provided invaluable experience in designing, implementing, and deploying distributed systems. The integration of a simulated sensor network and a custom front-end interface highlighted the complexity of real-world applications and allowed us to apply a full-stack development approach.

6.4 Skills gained

This course equipped me with both theoretical knowledge and practical skills necessary to design, deploy, and manage Service-Oriented Architectures (SOA). Specifically, I gained the following competences:

Skills	**Required**	**Achieved**
Software Engineering		
Define the different phases in software development	3	3
Know the different project management methods	3	3
Apply one of these methods to a project	3	3

Table II..7: Skill matrix for Software Engineering

Skills	**Required**	**Achieved**
Service Oriented Architecture		
Know how to define a Service Oriented Architecture	4	4
Deploy an SOA with web services	4	4
Deploy and configure an SOA using SOAP	4	4
Deploy and configure an SOA using REST	4	4
Integrate a process manager in an SOA	4	4

Table II..8: Skill matrix for Service Oriented Architecture

Competence	Description	Level of Mastery
Defining a Service-Oriented Architecture	Ability to identify and design modular, loosely coupled services that promote reusability and scalability.	Advanced
Deploying an SOA with Web Services	Practical experience implementing SOA with web services to enable interoperability across platforms.	Advanced
Configuring SOA with SOAP	Skills in deploying SOAP-based services, including WSDL design, handling SOAP requests, and integration.	Intermediate
Configuring SOA with REST	Proficiency in creating RESTful services using HTTP methods and lightweight data formats like JSON and XML.	Advanced
Integrating a Process Manager in SOA	Knowledge of service orchestration and choreography to coordinate workflows across multiple services.	Intermediate

Table II..9: Competences Gained During the Service-Oriented Architecture Course

6.5 Analysis and remarks

The Service-Oriented Architecture course was both challenging and rewarding. It gave me a solid introduction to SOAP, REST, and microservices, and helped me understand how these architectures are used in real-world software development. The combination of tutorials and projects made the learning process engaging, but it also required a lot of effort and time management. The tutorials, while very informative, were particularly time-consuming. They required additional work at home to ensure I could complete the exercises and still have time for the projects.
Coming from the Automatic Electronics speciality, and even though it was demanding, I found the tutorials really necessary. They covered key concepts such as service discovery and load balancing, which were crucial to the realization of the projects.

The two projects were where I really applied what I learned. In the first project, designing RESTful microservices for a volunteering application helped me grasp how to build APIs and manage service interactions. The second project, where we simulated a smart building, was even more complex. It was rewarding to see how sensors, actuators, and services could work together in a distributed system. It also showed me the challenges of coordinating multiple services and maintaining a functional system. I found the comparison between SOAP and REST very useful. SOAP is highly reliable and standardized, but REST is much simpler and better suited for most web applications. Learning about microservices was exciting too, as it showed how small, independent services can make systems more flexible and scalable. At the same time, I realized the extra complexity they add, especially when managing multiple services.

Overall, the course was intense but worthwhile. The combination of tutorials and projects gave me both theoretical knowledge and practical skills. It wasn't always easy to balance the workload, but the experience was definitely valuable and has prepared me to tackle real-world challenges in software development.

7. Wireless Sensor Networks

7.1 Context

Wireless Sensor Networks (WSNs) play an essential role in the development of intelligent, interconnected systems known as the Internet of Things (IoT). Unlike traditional networks, WSNs prioritize energy efficiency, robustness and scalability due to the limited nature of their hardware and the harsh environments in which they often operate.

The course was taught by the Professor Daniela Dragomirescu, and emphasizes the importance of balancing cost, energy efficiency, lifetime, and ease of deployment when designing such systems. It introduces me to the trade-offs involved in protocol design, from the physical layer to the application one, with a focus on practical deployment strategies and performance optimization.
The second part of the course followed a different approach, where students were tasked with presenting a detailed analysis of specific protocols for WSNs. My group focused on LoRa, while other groups presented on Sigfox, BLE, ZigBee, NB-IoT, and M2M protocols.

7.2 Technical Summary

In this course I discovered the unique specifications of WSN protocols such as Zigbee and Bluetooth Low Energy (BLE), and how they differ from the traditional networks. One of the key focus was on the critical requirement for low power consumption, as many WSN nodes are in a restricted energy environment. One of the most common energy-saving strategies involves inactive and sleep period time.

In the second assignment, we explored various Medium Access Control (MAC) layer protocols, which play a vital role in WSN efficiency. We can broadly classified them into three categories:

- **Contention-based protocols:** Nodes transmit data when the communication medium is free. This approach, often implemented using Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), is simple but can lead to collisions in high-traffic scenarios.
- **Scheduled-based protocols:** Nodes are assigned specific time slots for transmission, ensuring collision-free communication. Time Division Multiple Access (TDMA) is a common example, providing guaranteed bandwidth but requiring precise synchronization.
- **Hybrid protocols:** These combine features of both contention-based and scheduled-based approaches. For example, Zigbee uses a hybrid MAC layer, with part of its superframe dedicated to contention-based access and another part reserved for scheduled transmissions.

7.3 Practical Work

As part of this course, I worked on a project where my group and I designed a protocol for a real-world application in a constrained environment. In a specific topic: public sewer pipes. The objective was to deploy multiple sensors along the sewer pipelines to monitor their conditions. This challenging environment required a protocol that could meet several specific constraints:

- **Low energy consumption:** Essential for ensuring long-term operation.
- **Very low bandwidth:** To handle sparse data transmission efficiently.
- **Multi-hop communication:** Each sensor would act as a relay to transmit data further down the network.

For this project, we developed a new MAC protocol called 3N-MAC (Near Node Network for Wireless Sensor Networks). My role in the project focused on the implementation of the physical layer, with my teamates Paul Jaulhiac and Cyril Vasseur. We used GNU Radio to implement and test the physical layer, ensuring robust and reliable signal transmission despite the challenging environment. This hands-on experience allowed me to understand how the physical and MAC layers interact and the complexities involved in designing a system that balances energy efficiency, reliability, and performance.

7.4 Skills acquired

Through this course, its research part, and the labs, I consider to have acquired the following skills:

Skill	Description	Level of Mastery
Protocol Analysis and Design	Understanding and evaluating the trade-offs between different WSN MAC protocols to design networks tailored to specific application requirements. (Being able to suggest optimal technological solutions for IoT networks)	Advanced
Simulation and Evaluation	Using simulation tools to model and assess network performance under diverse environmental and operational constraints.	Intermediate
Energy Optimization	Developing strategies to balance energy efficiency and communication reliability in WSNs.	Advanced
Critical Comparison	Comparing and selecting the most suitable protocols for various IoT applications based on factors such as energy efficiency, scalability, and reliability. (Being able to analyze and evaluate optimal wireless network technologies)	Advanced
Problem Solving in Constrained Environments	Designing practical solutions for challenging real-world scenarios, such as our sewer pipe monitoring project.	Intermediate

Table II..10: Wireless Sensor Networks Skills

7.5 Analysis and remarks

Throughout this course, I realized how critical it is to select the right protocol when designing a Wireless Sensor Network (WSN). Each protocol has its own strengths and trade-offs, making it essential to align the choice with the specific requirements of the application. For example, protocols like S-MAC and T-MAC are excellent for conserving energy, but their reliance on synchronization and inability to handle mobility well make them better suited for static networks with moderate traffic. On the other hand, Z-MAC stood out for its hybrid approach, which balances energy efficiency and throughput by adapting to traffic levels. However, I noticed that Z-MAC's reliance on global synchronization can be a challenge in highly dynamic environments.

Working on the sewer pipe monitoring project helped me understand the practical challenges of deploying WSNs in constrained environments. While designing the 3N-MAC protocol, I gained hands-on experience in tailoring solutions to meet the specific demands of the application. However, the lack of detailed instruction on GNU Radio implementation hindered our progress in building a fully functional physical layer. I think that training on tools such as GNU Radio upstream of the project would have enabled us to be much more efficient.

Another important takeaway is that no single protocol is universally ideal for all WSN applications. For instance, energy-efficient protocols like B-MAC are great for low-traffic applications but may introduce delays or lack synchronization features. Similarly, location-based protocols like GAF are excellent for dynamic networks but rely on accurate localization, which can complicate deployment.

Overall, this course has deepened my understanding of the complexities involved in WSN design and deployment. It highlighted the importance of making informed trade-offs based on the specific needs of the application and provided me with a strong foundation to approach similar challenges in the future. The hands-on project reinforced the theoretical knowledge, but with more practical training on software tools, I believe I could have gained even more from the experience.

8. Middleware for IoT

8.1 Context

The Middleware for IoT course was taught by Professor Thierry Monteil. The main objectives of this course were to explore different approaches and communication protocols, and to understand the challenges of communication between connected devices in order to implement solutions tailored to specific needs.

Like the course Service Architecture taught by Ms. Nawal Guermouche, this course was delivered in the form of a MOOC (Massive Open Online Course). Afterwards, we carried out a series of practical exercises designed to illustrate the concepts discussed in the MOOC, notably the use of MQTT, oneM2M, and Node-RED.

The goal of these practical sessions was to understand how multiple connected devices can interact with each other by choosing appropriate protocols, while taking into account the variety of existing hardware and software infrastructures.

8.2 Technical Summary

As mentioned earlier, we studied several communication protocols during this course, particularly Message Queuing Telemetry Transport (MQTT). This is a lightweight messaging protocol designed for devices with limited resources. Its functioning relies on a broker (a server) that manages publication topics. Client publishers publish messages on specific topics, and client subscribers receive messages by subscribing to these topics. Thanks to its simple architecture, the MQTT protocol is widely used in IoT, where bandwidth consumption must be minimized and reliability is essential.

The second protocol we studied is oneM2M. More precisely, it is a standard endorsed by multiple standardization bodies, aiming to provide and ensure interoperability among IoT architectures. Based on REST, oneM2M proposes a hierarchical structure to model resources for an IoT application. The resources are organized on three levels: Application Entities (AE), Containers (CNT), and Content Instances (CI). These levels define the hierarchical storage space on servers. Application Entities (AE) represent available functionalities, within which there are Containers (CNT) that serve as data categories and Content Instances (CI) that hold published data instances. Thus, it is through a container in an AE that an IoT device can publish or read information.

8.3 Practical Work

The first project in the practical sessions involved deploying an MQTT broker using Mosquitto and setting up simple message exchanges among several ESP8266 devices. One ESP8266 acted as a button (the publisher), while another controlled an LED (the subscriber). Any change in the button's state was reflected by turning the LED on or off.

The second project, which was essentially the same as the first but with oneM2M, began with following a tutorial in a Jupyter Notebook to understand the core concepts. The first exercises in the notebook enabled us to learn how to create AEs, containers, and content instances. We then went a step further by exploring group creation, notifications, and rules that can automatically update resources.

Only after mastering these concepts did we use a oneM2M server and a Python script to simulate the same application we had built with MQTT. To achieve an equivalent setup to the first project, my partner and I simulated a virtual "button" and a virtual "LED." A Python script allowed us to create AEs, containers, and content instances for both the button and the LED, as well as manage notifications.

Had we had more time for this project, I would have liked to integrate Node-RED to more easily manage data flows among the various IoT devices. Node-RED would have provided a visual interface to intuitively connect hardware devices, APIs, and online services.

8.4 Skills Acquired

Skills	Required	Achieved
Middleware for the IoT		
Know how to situate the main standards for the Internet of Things	4	4
Deploy an architecture compliant to an IoT standard and implement a sensor network	4	4
Deploy and configure an IoT architecture using OM2M	4	4
Interact with the different resources of the architecture using REST services	4	4
Integrate a new technology into the deployed architecture	4	4

Table II..11: Skill matrix for Middleware for the IoT

Skill	Description	Level of Mastery
Situate IoT Standards	Know how to position the main standards for the Internet of Things	Advanced
Deploy IoT Architecture	Deploy an architecture compliant with an IoT standard and implement a sensor network	Intermediate
Configure OM2M	Deploy and configure an IoT architecture using OM2M	Intermediate
Interact with REST	Interact with different resources of the architecture using REST services	Advanced
Integrate New Technology	Integrate a new technology into the deployed architecture	Intermediate

Table II..12: Skills Acquired During the Middleware for IoT Course

8.5 Analysis and Remarks

Discovering and experimenting with two complementary IoT protocols—MQTT (lightweight, suitable for quick use cases with minimal resource consumption) and oneM2M (more structured and standardized, offering a comprehensive framework for large-scale IoT applications) has helped me understand the importance of interoperability in the Internet of Things. The practical work was especially instructive, from configuring a Mosquitto broker and designing a mini-project on ESP8266 devices, to exploring oneM2M via Python scripts and using Node-RED for data flow management. However, I would have liked to delve deeper into the direct integration of oneM2M into Node-RED to fully leverage its potential, particularly in terms of notifications and management rules.

In addition, installing the necessary tools on our personal machines slowed down my partner and me, especially for the oneM2M tutorial in the Jupyter Notebook. I would have appreciated having a ready-to-use solution such as a virtual machine to deploy at the start of the practical sessions, similar to what was provided for the Service Architecture TP sessions.

From a personal standpoint, this course enhanced my academic path by giving me a more comprehensive view of different communication approaches in IoT. The skills I acquired will be valuable for future projects that require choosing the most appropriate solution for a specific context. Intending to pursue the Master's Program in TLS-Sec after completing my INSA degree, I regret that the security aspects of these protocols were not addressed during the practical work. Ensuring secure exchanges (encryption, authentication, intrusion protection) is a major challenge in IoT.

9. Security for Connected Objects

9.1 Context

The course "Security for Connected Objects" was delivered by E. Alata and V. Migliore and focused on specific security issues, as its name suggests, related to IoT devices. We explored theoretical and practical concepts, security protocols, cryptographic techniques, and vulnerability studies.
In an era where connected objects are ubiquitous, this course highlighted the unique security challenges of IoT systems, driven by their resource, performance, and connectivity constraints.

9.2 Technical Summary

During Mr. Alata's classes, we addressed the security of web applications, secure communication protocols, and microarchitectures. Web application security had already been covered in a previous 3A course with the professor, particularly with the introduction of SQL injection attacks, though not as in-depth as this time.

Web Application Security

The lectures and associated practical work on web application security enabled us to understand common vulnerabilities in web applications and techniques to exploit them. Specifically, we studied SQL injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF) attacks, as well as the protective measures implemented to counter them. We also covered session and cookie management issues and best practices for securing web applications, such as input validation, output escaping, and implementing access controls.

Security of Communication Protocols

In this part of the course, we revisited the "fundamental" security properties, including confidentiality, ensuring that exchanged information is inaccessible to third parties; integrity, ensuring that data has not been altered; authenticity, allowing the verification of the identity of communicating entities; and non-repudiation, preventing entities from denying their participation in communication. We also discussed the concept of "post-compromise security," which aims to ensure that even after a key is compromised, future communications can regain security.

We reviewed concrete examples such as replay attack protection using nonces or timestamps. This section also introduced us to tools like ProVerif for formalizing protocols and assessing their security against various types of attacks.

Microarchitecture Security

This section, the shortest of the three, addressed hardware attacks exploiting the microarchitecture of components. Examples included Spectre and Meltdown attacks, which leverage microprocessor branch prediction to execute instructions accessing sensitive data. Reverse engineering was also discussed, allowing for an understanding of the internal microarchitecture of processors through experimentation. For instance, the size of a hidden buffer can be determined by measuring memory access times.

Finally, we examined vulnerabilities related to instruction pipelines, including manipulation of RAW (Read After Write) hazards.

Mr. Migliore's classes covered two main areas: applied cryptography and hardware security. These two complementary dimensions helped us understand the vulnerabilities and appropriate solutions for IoT systems, where resource constraints and the diversity of threats make securing them particularly complex.

Applied Cryptography

In this module, we explored the fundamentals and modern applications of cryptography. The course began with a review of basic notions, including substitution and permutation principles, with examples like Caesar, Vigenère, and Hill ciphers. Modern cryptography was then emphasized, covering security properties (confidentiality, integrity, authenticity, non-repudiation) and symmetric (AES) and asymmetric (RSA) encryption algorithms. Particular attention was paid to block cipher modes of operation, such as Cipher Block Chaining and Counter Mode, along with their security implications.

We briefly addressed a lightweight cryptography standard, ISO/IEC 29192, specifying algorithms tailored to the limited resources of connected objects.

Hardware Security

This part of the course served as a refresher on hardware security topics previously covered in 3A. We revisited physical attacks, such as side-channel and fault attacks, along with their countermeasures. Specifically, we examined power consumption analysis attacks, cache timing attacks, and RowHammer attacks. Since we had already conducted practical work on these attacks in prior years, the professor deemed it unnecessary to repeat the same exercises.

9.3 Practical Work

The practical work conducted during the course provided an opportunity to apply theoretical concepts studied in lectures. These activities deepened my understanding of security techniques for connected objects and developed my practical skills.

SQL Injection Attacks

We experimented with SQL injection attacks on a vulnerable database. Using malicious queries, we demonstrated how to bypass authentication systems and access sensitive data. To secure these systems, we implemented parameterized queries and stored procedures to prevent the execution of malicious commands.

Cross-Site Scripting (XSS)

We simulated XSS attacks where a malicious script is injected into a web application. One of the first steps in the lab involved executing JavaScript code in a user input field, allowing us to display a message in a user's browser. This lab allowed us to explore protective measures such as escaping special characters and implementing security headers like Content-Security-Policy.

Man-in-the-Middle (MITM) Attacks

In this lab, we used the mbedTLS library to implement a MITM attack, where an attacker intercepts and modifies communications between two parties. A simulation demonstrated how an intercepted certificate could be altered, compromising the confidentiality and integrity of exchanges. This work highlighted the importance of valid certificates and verifying "Common Name" fields to prevent such attacks.

Quantum Manipulation

An innovative lab explored the principles of quantum communication, particularly quantum key distribution (QKD). We configured a system using polarized photons to share a cryptographic key, focusing on equipment calibration and the robustness of communications against attacks.

Innovative Project - Wispers

As part of the Innovative Project, we implemented the RuBee protocol, a low-frequency wireless communication protocol, which required us to establish a "Frame Check Sequence" (FCS). The FCS, designed to verify the integrity of frame data, was encoded on one byte and calculated by dividing (modulo 2) the product of x^8 multiplied by the frame content by the generator polynomial $x^8 + x^2 + x + 1$.

Lacking information on the reliability of this FCS in the IEEE standard, we conducted fuzzing tests to assess its robustness. Assuming that only one FCS should exist for a given frame size, we determined it was feasible to generate different frames with the same FCS. However, since the FCS is meant to detect bit value changes due to interference, we tested its robustness by modifying one bit in a given frame size and verifying if the FCS detected the modification.

After 9 hours of computation, we were able to scale up to frames of 9 words of 4 bits without detecting collisions.
You can find our scripts in the "Python" folder of our Wispers project repository at `https://github.com/NoNo47400/WispersProject`.

9.4 Skills Acquired

Skills	Required	Achieved
Security for connected objects		
Knowing the main issues in security for IoT	3	3
Understand the terminology of security	2	2
Being able to have a critical look at the design of a system from a security point of view	2	2
Being able to understand a scientific article that explains a weakness or a security solution and to explain it	2	2

Table II..13: Skill matrix for Security for Connected Objects

Skill	Description	Level of Mastery
Understanding IoT security issues	Ability to identify and understand security challenges in IoT systems, including constraints like resource limitations and connectivity vulnerabilities.	3/4
Knowledge of cryptographic techniques	Familiarity with modern cryptographic algorithms such as AES, RSA, and lightweight standards like ISO/IEC 29192, as well as their applications in securing IoT.	3/4
Practical application of security protocols	Experience in implementing and verifying security protocols using tools like ProVerif and programming libraries (e.g., mbedTLS).	3/4
Understanding and mitigating web vulnerabilities	Proficiency in identifying vulnerabilities such as SQL injection, XSS, and CSRF, and implementing countermeasures to secure web applications.	3/4
Analyzing hardware security threats	Knowledge of physical and side-channel attacks (e.g., cache timing, RowHammer, Spectre) and their countermeasures.	2/4
Use of quantum principles in security	Basic understanding of quantum communication and key distribution mechanisms to enhance security in IoT systems.	2/4
System design analysis	Ability to critically evaluate the design of IoT systems from a security perspective and propose improvements.	3/4

Table II..14: Skill matrix for Security for Connected Objects

9.5 Analysis and Remarks

The course "Security for Connected Objects" was an extremely enriching experience. It allowed me to better understand the unique challenges of securing IoT systems by combining theoretical and practical aspects. Mr. Alata's lectures deepened my understanding of secure communication protocols and web vulnerabilities, while Mr. Migliore's classes enhanced my mastery of cryptographic foundations and hardware attacks.

The practical aspect of the course, particularly the work on AES and Man-in-the-Middle attacks, provided a clear overview of real-world challenges in connected systems. These activities strengthened my ability to analyze security architectures, think of robust solutions, and understand the complex interactions between hardware and software.

This course solidified my decision to pursue the TLS-SEC Specialized Master's (Toulouse Security) next year. It helped me develop key skills essential for my future career and motivated me to further my studies in the field of computer security. I am also confident that the security concepts covered in this course will have direct applications in the coming months as I join the security team at Schaeffler.

10. Energy for connected objects

10.1 Context

The “Energies for Connected Objects” course was given by Gaël Loubet. In this course, we explored possible existing solutions for powering IoT devices without batteries or wired systems. Once the basics had been laid down, the course focused on ambient energy harvesting, wireless energy transfer and various solutions for making systems energy autonomous.

This course was given in parallel with the course on wireless sensor networks, in which the issue of energy consumption was also highlighted as a decisive factor in the choice of protocol definition.

10.2 Technical Summary

It contained discussions on capacitors and supercapacators to buffer energy, wireless power transfer that has been highly evolved with a near-field mechanism by means of capacitive and inductive coupling to a far-field radiative by the use of a rectenna, and energy harvesting techniques using ambient energy through conversion of energy due to light, mechanical motion, thermal gradient, and electromagnetic waves. The course also integrated optimization strategies, such as antenna design to match the frequency and software optimizations that would reduce energy consumption in connected devices.

10.3 Practical work

The practical work for this course involved the design and testing of energy systems in the laboratory. One of the projects was to power an LED from ambient RF energy. This involved analyzing a rectifier circuit, choosing appropriate antennas, and optimizing the energy transfer for efficient operation. I used GNURadio and spectrum analyzers to measure RF power output, find the optimal operating frequencies, and test various energy storage configurations. The "store then use" strategy, as realized with a bq25504 power management unit and a TPS63031 DC-DC converter, allowed for efficient energy buffering and utilization in low-power scenarios.

10.4 Skills acquired

In this course, I developed high-level technical expertise in low-power circuit design, energy harvesting systems, and wireless power transfer technologies. I acquired hands-on experience with laboratory tools and methodologies such as frequency sweeps, impedance matching, and antenna characterization. Beyond the technical skills, I learned to interpret complex system data and optimize designs for practical applications. The course also fostered innovative problem-solving skills, particularly in balancing theoretical frameworks with real-world constraints.

Skills	Required	Achieved
Energy for Connected Objects		
Understand and master optimisation of communication protocols for IoT with respect to energy limitations	4	4
Mastering the architecture of an energy management system, simple storage, energy recovery, know how to size the storage element according to the specifications	4	4

Table II..15: Skill matrix for Protocols and Communication

10.5 Analysis and remarks

This class demonstrated the potentials and pitfalls of designing energy-autonomous IoT devices. Even though I implemented various systems, such as the energy harvesting LED with ease, environmental inputs had high variability that posed the biggest challenges. For instance, supplying power continuously in constantly fluctuating electromagnetic environments forced me to devise creative solutions but also highlighted the necessity of using hybrid energy systems. Furthermore, a central design challenge was determining the balance between optimizing the capture of energy and reliability of the system.

10.6 Reflections

"Energy for Connected Objects" provided a great platform for innovative energy solution development within the IoT domain. It identified the key challenges arising from future technologies in relevance to sustainability and efficiency; therefore, this was focused on the latest energy system approaches and applications. In the future, my skills and knowledge will guide my work in energy-autonomous devices, mainly in scalable wireless sensor networks, but also including advanced material development for energy conversion and storage. These experiences crystallized my aspiration to further the frontier of IoT energy systems toward sustainable technological development.

11. LAB at AIME

11.1 Context

INSA Toulouse has an Interdisciplinary Microelectronics and Electronics Workshop (AIME), which is a research and training laboratory in microelectronics and electronics. As part of the ISS program, we spent a week there to build a gas sensor through the chemical synthesis of WO_3 nanoparticles and their integration into microelectronic devices. This lab is equipped with clean rooms that allow high-precision procedures, essential for micro-scale fabrication.

However, the initial fabrication steps, up to photolithography step 2 (“Contacts Opening”), had already been carried out by the lab technicians, allowing us to focus on the subsequent stages.

11.2 Technical Summary

The project aimed to produce a gas sensor based on WO_3 nanoparticles, a semiconductor material known for its gas sensitivity. The main steps were, in order:

1. Carry out photolithography step 3 for the aluminum electrode etching.
2. Assemble the devices, including cutting and mounting the chips on TO5 supports.
3. Synthesize WO_3 nanoparticles, involving seed preparation and nanowire growth via hydrothermal processes.
4. Integrate WO_3 nanorods onto interdigitated combs using dielectrophoresis.
5. Perform electrical characterization of the sensors in a controlled atmosphere containing ethanol or ammonia.

11.3 Practical Work

During the week, we were able to complete steps 1 through 5 of the lab. We worked in groups of four, and each group was responsible for its own sensor. I was in charge of synthesizing the WO_3 nanoparticles and integrating them onto the interdigitated combs. I also participated in the electrical characterization of the sensors.

11.4 Skills Acquired

Thanks to this lab, I believe I have acquired the following skills:

Skill	Description	Level of Mastery
Understanding Basic Concepts of Sensors and Data Acquisition	Understanding the fundamental principles of sensors, physics, electronics, and metrology for data acquisition.	Advanced
Fabrication of Nanoparticle-based Sensors	Being able to build a gas sensor using microelectronics tools: chemical synthesis, integration, and testing.	Intermediate
Using Microelectronics Tools in Clean Rooms	Following protocols and handling equipment for steps such as photolithography, metallization, and nanoparticle integration.	Advanced
Creating a Datasheet	Knowing how to write a technical datasheet for a gas sensor, including electrical characteristics and test conditions.	Intermediate

Table II..16: Skills acquired during the week at the AIME laboratory

Not coming from a physicist or chemist background, I can't say I've completely mastered the subject. However, after this week in the lab, I feel capable of following a fabrication protocol and characterizing a sensor.

11.5 Analysis and remarks

As a work-study student, this week at AIME was a real discovery for me of the research environment, and more specifically of microelectronics. I was able to explore and handle various steps in the fabrication of microelectronic devices, ranging from photolithography to nanoparticle integration.

Working in groups of four also allowed me to understand teamwork in a research context, by sharing tasks and collaborating to achieve a common goal. I believe this experience has helped me to better grasp the challenges of research as well as the importance of following a strict experimental protocol without making errors during the procedures.

12. Microcontrollers, Open-Source Hardware, and Sensor Introduction

12.1 Context

After a week at AIME dedicated to building gas sensors, we attended classes in the physics department. These classes focused on creating the sensor's technical datasheet and sizing the components for the adaptation stage that reads the sensor values. We also received an introduction to microcontrollers, using KiCad for PCB design, and the operation of ChirpStack and NodeRed. The goal of this module was to teach us how to build a complete system from the gas sensor and its datasheet to a web user interface in NodeRed.

12.2 Technical Summary

Spanning several weeks, this module began with the gas sensor development week at AIME. Afterwards, we had an introduction to microcontrollers, primarily aimed at students from the IR (Computer Engineering) track, who discovered microcontrollers using Arduino boards.

Since my teammate and I come from the AE speciality and were more comfortable with microcontrollers, we chose to work with an ESP8266 and connect it to INSA's LoRa network using an RN2483 module. However, this module proved to be barely functional, likely due to power supply errors made by students in previous years. Indeed, it is supposed to be powered at 3.3 V, but Arduino signals are 5 V, which damages the module in the absence of a level-shifting stage.

Faced with these issues, I decided to use my own ESP32 modules, which integrate a LoRa module directly. We also encountered problems with INSA's ChirpStack gateway, which was often saturated, allowing us to send frames successfully only once every ten attempts. I therefore decided to use my own gateway, which is registered on The Things Network (TTN) and worked perfectly. Once the LoRa frames were visible on the TTN website, we used an MQ-3B sensor to simulate our gas sensor's operation.

Next, we took a component-sizing course using LTSPICE, to understand our adaptation stage and correctly dimension the components. We also had an introduction to Git. This was not new to me, since I use Git in my company, but it allowed my classmates to discover this version-control tool. We then learned about MIT App Inventor, a tool for quickly building Android apps. By adding a Bluetooth module to our ESP32, we were able to control an LED from the app.

We then moved on to NodeRed, a block-based programming tool that makes it very easy to create web applications. We used it to build a web interface for viewing our gas-sensor data, retrieving these data via MQTT from the TTN platform.

Finally, we were introduced to KiCad, a PCB design software. We used it to create a "shield" for our ESP32, incorporating the adaptation stage for the gas sensor. Due to time constraints, I ordered the PCB from JLCPCB and assembled the components upon its arrival.
You can find the entire project on my GitHub: https://github.com/Raspeur/AIME_

sensor

12.3 Skills Acquired

Skills	**Required**	**Achieved**
Introduction to Sensors		
Understand basic notions of sensors, data acquisition: physics, electronics and metrology point of view	4	4
Be able to manufacture a nano-particles sensor using micro-electronics tools: chemical synthesis, assembly, testing	4	4
Be able to design the datasheet of the sensor manufactured	4	4

Table II..17: Skill matrix for Introduction to Sensors

Skills	**Required**	**Achieved**
Microcontrollers and Open Source Hardware		
Understand microcontroller architecture and how to use them	4	4
Be able to design data acquisition system (sensor, conditioner, microcontroller) with respect to the application	4	4
Be able to design the electronic circuit of a sensor's signal conditioner (design + simulation)	4	4
Be able to design a shield to accommodate the gas sensor	4	4
Be able to design the software to use the gas sensor and its HMI	3	3
Be able to combine all of the above mentioned components into a smart device	4	4

Table II..18: Skill matrix for Microcontrollers and Open Source Hardware

Skill	Description	Level of Mastery
Understand Microcontroller Architecture	Understanding microcontroller architectures, functionalities, and use cases.	Advanced
Data Acquisition System Design	Designing end-to-end data acquisition systems, including sensor selection, signal conditioning, and microcontroller integration.	Advanced
Sensor Signal Conditioner Circuit	Developing and simulating signal conditioning circuits to ensure accurate data acquisition.	Advanced
Shield Design for Gas Sensor	Designing and fabricating custom hardware expansions (shields) for proper integration of gas sensors.	Advanced
Sensor Software and HMI Development	Implementing embedded software solutions, including sensor drivers, data processing, and user interfaces for gas sensing applications.	Advanced
Smart Device Integration	Combining hardware and software components into a cohesive, fully operational IoT device.	Advanced

Table II..19: Skills Acquired During the Microcontrollers and Open-Source Hardware Module

12.4 Analysis and remarks

This course was very interesting to me because it allowed me to discover new tools such as KiCad and NodeRed, and to use them in an end-to-end project. We went from creating a sensor all the way to a web interface for data visualization, including designing a PCB. It was very rewarding to see the project come to life and to manage our time effectively to complete everything on schedule.

I also enjoyed helping classmates from the IR track, who did not have much experience with microcontrollers and were not always very comfortable with them.

As part of this course, I also learned about licensing concepts whether open source or proprietary and all their implications. This is a notion I consider very important, and it was the first time I had addressed it in my studies.

Finally, I got to learn more about the functioning and limitations of INSA's ChirpStack gateway, which prompted me to use my own LoRa gateway registered on the LoRaWAN TTN network.

13. Communication protocols for LP-WPAN

13.1 Context

This module provided an in-depth understanding of low-power, short-range wireless communication technologies specifically designed for constrained IoT devices. The focus was primarily on IEEE 802.15.4, a foundational standard for several IoT protocols, including Zigbee and 6LoWPAN, enabling efficient and reliable communication in resource-limited environments.

Additionally, the course emphasized the integration of IPv6 in IoT networks, highlighting its vast benefits such as scalability, address space expansion, and compatibility with IoT-specific protocols. The practical sessions and theoretical discussions explored IPv6 implementation challenges and solutions, particularly its adaptation for constrained networks using 6LoWPAN and routing protocols like RPL. This dual focus on LP-WPAN and IPv6 provided a comprehensive perspective on modern IoT communication architectures.

13.2 Technical Summary

This course focused on the design and implementation of IPv6-based communications in IoT networks, emphasizing its adaptation to constrained environments such as LP-WPAN. The course began by exploring the unique characteristics of IoT networks, such as scalability, energy efficiency, and reliance on specialized protocols like IEEE 802.15.4. These characteristics set IoT networks apart from traditional computer networks and require tailored solutions for low-power devices operating under dynamic and often lossy conditions.

The adoption of IPv6 in IoT systems was a key focus. IPv6 provides a vast addressing space, supporting billions of devices with unique addresses and enabling direct end-to-end communication without the need for NAT. Practical lab sessions demonstrated the steps involved in initializing IPv6, including address auto-configuration using MAC addresses, the Duplicate Address Detection (DAD) process, and the Neighbor Discovery Protocol (NDP). These mechanisms are crucial for the dynamic integration of nodes into IoT networks.

To limit and even reduce IPv6 header overhead in resource-constrained networks, the course introduced the 6LoWPAN protocol. This protocol compresses IPv6 headers, significantly reducing data size to fit the limited frame sizes of IEEE 802.15.4. Additionally, the RPL (Routing Protocol for Low-Power and Lossy Networks) was studied as a solution for efficient multi-hop routing. The combination of 6LoWPAN and RPL, within the IETF standardized IoT protocol stack, ensures energy-efficient, scalable, and reliable communication in IoT networks.

The course also examined existing IPv6-based IoT technologies, including Thread for smart homes, LTE-M for large-scale mobility, and Zigbee IP for building automation. Each technology demonstrated the practical application of IPv6 in real-world IoT scenarios. These discussions highlighted the relevance of IPv6 for modern IoT applications, particularly its potential to enhance scalability and modularity in projects like Wispers, even when faced with low-frequency transmission constraints.

13.3 Practical Work

The practical sessions provided hands-on experience with LP-WPAN technologies, focusing on configuring IPv6 and 6LoWPAN. The practical sessions for the LP-WPAN module were structured around two interconnected labs, focusing on configuring and analyzing IPv6 networks and exploring 6LoWPAN and RPL protocols. These labs were designed to provide hands-on experience with the technologies discussed in the theoretical component of the course, enabling the application of concepts in simulated IoT environments.

In the first lab, the focus was on understanding IPv6 auto-configuration processes. This included tasks such as assigning link-local and global addresses, performing Duplicate Address Detection (DAD), and analyzing the behavior of the Neighbor Discovery Protocol (NDP). Using tools like Wireshark, we captured and studied IPv6 packets, including router advertisements and multicast traffic. These activities not only demonstrated the initialization steps of IPv6 but also provided insights into how IPv6 ensures scalability and compatibility in IoT systems.

The second lab expanded on this foundation by delving into 6LoWPAN header compression and the RPL (Routing Protocol for Low-power and Lossy Networks). We utilized Mininet-WiFi to simulate a constrained IoT network and experimented with 6LoWPAN's ability to reduce IPv6 overhead, improving data transmission efficiency over IEEE 802.15.4. Additionally, we explored RPL's functionality in building and maintaining routing paths in multi-hop networks. By examining DODAG formations and routing tables, we gained a practical understanding of how RPL supports reliable communication in low-power environments.

13.4 Skills acquired

Competence	Description	Level of Mastery
IPv6 Network Configuration	Setting up IPv6 with auto-configuration and NDP for IoT networks.	Advanced
6LoWPAN Header Compression	Reducing IPv6 header sizes to improve efficiency in LP-WPAN.	Advanced
RPL Implementation	Configuring and analyzing RPL routing in multi-hop IoT networks.	Intermediate
Wireshark Analysis	Capturing and interpreting IPv6 traffic to identify optimization opportunities.	Advanced
IoT Sustainability Design	Balancing network performance with energy efficiency.	Intermediate

Table II..20: Competences Gained During the LP-WPAN Module

13.5 Analysis and remarks

This module underscored the challenges and innovations of constrained IoT networks. The guided tutorials and simulation project provided a hands-on understanding of IPv6 adaptation for LP-WPAN, particularly through 6LoWPAN and RPL. The experience revealed the critical trade-offs between performance, power consumption, and scalability in IoT designs. This learning experience not only expanded my technical skillset but also offered valuable insights into the sustainability implications of IoT technologies.

14. Embedded AI for IoT

14.1 Context

The module "Embedded AI for IoT," taught by Professor Philippe Leleux, builds upon an initial introduction to machine learning that we took during the fourth year. This year, the course focused on integrating artificial intelligence (AI) into embedded systems. The aim was to provide us with a comprehensive understanding of the challenges and opportunities presented by the combination of AI and the Internet of Things (IoT). The primary objective was to explore how AI can enhance the capabilities of embedded systems, enabling them to perform complex tasks autonomously.

14.2 Technical Summary

In this course, we covered essential theoretical and practical concepts related to integrating AI into constrained devices. We studied data preprocessing and feature engineering, which included dimensionality reduction techniques such as Principal Component Analysis (PCA) and feature selection.
We also explored methods for data analysis and classification, focusing on algorithms like Dynamic Time Warping (DTW) and Recurrent Neural Networks (RNNs).
The course delved into deep learning, particularly Convolutional Neural Networks (CNNs), which are well-suited for resource-limited environments.
We discussed distributed learning paradigms, enabling AI models to be trained on devices while preserving data privacy. The trade-offs between cloud-based AI, edge AI, and on-device AI in terms of bandwidth, latency, and reliability were also highlighted.

14.3 Practical Work

We had two practical sessions for this course. The first session focused on recognizing and classifying human activities. The data was collected from three sensors (accelerometers) placed on the wrist, chest, and ankle of an individual. Using this data, we classified activities such as sitting, standing, walking, cycling, Nordic walking, vacuuming, and ironing. During this session, we implemented Dynamic Time Warping (DTW) to compare and align time-series data. We then classified the activities using k-Nearest Neighbors (k-NN) and applied dimensionality reduction with PCA. Finally, we implemented a neural network (multi-layer perceptron) for activity classification. The tools used in this session included Python, scikit-learn, and numpy.

The second practical session focused on fall detection using image data, emphasizing deep learning techniques. The dataset consisted of images capturing various fall scenarios. We built and trained a Convolutional Neural Network (CNN) to classify images into two categories: "fall" and "no fall." We applied transfer learning and model optimization techniques, such as quantization and pruning, to improve the model's efficiency (a key consideration for constrained embedded systems such as microcontrollers). The tools explored in this session included TensorFlow, Keras, and TensorFlow Lite.

14.4 Skills Acquired

Skills	Required	Achieved
Embedded AI for IoT		
Understand the characteristics of supervised and unsupervised learning problems	4	4
Understand the main basic methods and algorithms to deal with these problems	4	4
Understand the specificities of AI at the edge	4	4
Understand the main optimization methods enabling the embedding of AI algorithms	4	4
Be able to use these methods through Python libraries to solve practical problems with IoT data	4	4

Table II..21: Skills matrix for various software and data-related training units

Competence	Description	Level of Mastery
Data Preprocessing	Cleaning, scaling, and transforming IoT data for machine learning tasks.	Advanced
Time-Series Analysis	Classification and alignment techniques for temporal data from IoT sensors.	Intermediate
Deep Learning for IoT	Designing and optimizing CNNs for resource-constrained environments.	Advanced
Edge AI and Federated Learning	Implementing distributed learning paradigms while preserving privacy and improving performance.	Intermediate
TinyML and Model Optimization	Applying pruning, quantization, and transfer learning to create lightweight AI models.	Intermediate
Python Libraries	Proficiency with TensorFlow, TensorFlow Lite, scikit-learn, and Keras for embedded AI development.	Intermediate

Table II..22: Skills Acquired During the Embedded AI for IoT Module

14.5 Analysis and Remarks

This course was very engaging as it allowed me to better understand some of the concepts I had previously encountered during a LinkedIn Learning course by Professor Céline Robardet from INSA Lyon. The practical sessions, especially the fall detection project, were particularly rewarding. It was gratifying to see how techniques like CNN optimization and federated learning could be applied to solve real-world IoT problems.

15. Master REOC

15.1 Context

The Master REOC program is a 6 months program done in parallel of the 5ISS, the Thursdays afternoons. It is done in collaboration between INSA Toulouse and ENSEEIHT.

The objective of this program is to train network and telecommunications engineers, system/network architects, system/network administrators, and security engineers.

15.2 Technical summary

The Master REOC is structured to provide both theoretical knowledge on specific networking and telecommunications challenges, as well as practical skills to implement available solutions to address these issues.

The program consists of 10 distinct courses, each focusing on a specific aspect of networking and telecommunications.
We explored topics such as virtualization, quality of service (QoS), network modeling, network simulation, network security, wireless network performance, IoT middleware modeling, and LoRa network modeling. Additionally, the program covered more specialized subjects, such as integrated avionics modules and worst-case traversal time analysis in switched Ethernet networks.

15.3 Skills acquired

Through this course, I consider to have acquired the following skills:

Competence	Description	Level of Mastery
QoS Design	Designed networks with guaranteed Quality of Service for mission-critical applications.	Advanced
Network Constraint Analysis	Evaluated latency, throughput, and reliability constraints in embedded systems.	Intermediate

Table II..23: Skills Acquired in the "Embedded Networks and QoS" course

Competence	Description	Level of Mastery
Embedded System Virtualization	Implemented virtualization techniques to enhance resource utilization in embedded devices.	Advanced
System Partitioning	Applied partitioning principles to improve modularity and scalability.	Intermediate

Table II..24: Skills Acquired in the "Virtualization in Embedded Systems"course

Competence	Description	Level of Mastery
Wireless Network Optimization	Analyzed and optimized performance metrics like latency and throughput in wireless systems.	Advanced
Real-Time Performance Evaluation	Conducted real-time performance analysis for wireless networks in constrained environments.	Intermediate

Table II..25: Skills Acquired in the "Wireless Network Performance Evaluation" course

Competence	Description	Level of Mastery
ARINC 653 Principles	Applied modular design principles to Integrated Modular Avionics (IMA).	Advanced
Partitioning for Isolation	Designed systems with spatial and temporal isolation using ARINC 653.	Intermediate

Table II..26: Skills Acquired in the "Modular Architectures" course

Competence	Description	Level of Mastery
WCTT Calculation	Computed worst-case traversal time using network calculus for real-time Ethernet.	Intermediate
Latency Optimization	Optimized network latency to meet real-time constraints.	Advanced

Table II..27: Skills Acquired in the "Worst-Case Traversal Time Analysis" course

Competence	Description	Level of Mastery
IoT Middleware Design	Modeled middleware for efficient communication in IoT systems.	Advanced
LoRa Network Simulation	Simulated and evaluated LoRa networks for low-power IoT applications.	Advanced

Table II..28: Skills Acquired in the "IoT Middleware and LoRa Modeling" course

Competence	Description	Level of Mastery
Defining and Deploying Virtual Networks	Designing and implementing virtual networks, ensuring proper communication between components.	Advanced
Working with SDN and NFV	Gained hands-on experience creating VNFs, managing SDN controllers, and addressing network challenges.	Advanced
Network Monitoring and Optimization	Implemented systems to track bandwidth and mitigate congestion through SDN rule adjustments.	Advanced
Collaborative Problem-Solving	Successfully worked in a team to design and implement a complex network simulation.	Advanced

Table II..29: Competences Gained in the "Network Virtualization" (SDN & NFV) Course

15.4 Analysis and remarks

The Master REOC program was one of the most demanding experiences of my academic journey. The practical sessions, particularly the Mininet project, required intense focus and teamwork to complete within the constraints of the lab. While the workload was challenging, it was incredibly rewarding to see our theoretical knowledge applied to real-world scenarios.

This project helped me develop a deeper understanding of constrained networks and cutting-edge technologies like SDN and NFV. The process of designing and simulating a network topology taught me the importance of planning and precision, while implementing VNFs and developing the Flask interface allowed me to enhance my technical and problem-solving skills.

If I had the opportunity to redo this year, I would organise myself better. The challenges and time constraints, in parallel of my courses at INSA, were quite imposing, and pushed me to improve both my technical abilities and my ability to work under pressure. In fact, I think that this was a great experience, but I would have liked to have more time to explore the different courses.

Chapter III.

Conclusion

In this portfolio, I have detailed my academic and professional background, highlighting the technical and transversal skills I have acquired. My studies at INSA, focused on intelligent systems, have allowed me to explore various fields such as the Internet of Things (IoT), cybersecurity, artificial intelligence for embedded systems, energy optimization, and telecommunication networks.

The numerous projects I have completed this year, such as designing a nanotechnology sensor at AIME, developing embedded systems, and implementing communication protocols, have strengthened my technical skills and my ability to work in a team.

My international experience in Romania, as well as my collaborations with teams both locally and worldwide, have enabled me to develop skills in teamwork, project management, and intercultural communication.
Building on these experiences, I aim to further my studies by enrolling in the Mastère Spécialisé TLS-SEC (Toulouse Security). This program will allow me to deepen the knowledge I have acquired in security over the past few years and specialize in this rapidly expanding field.

This portfolio not only illustrates my academic journey but also showcases my technical and transversal skills, as well as my professional aspirations.