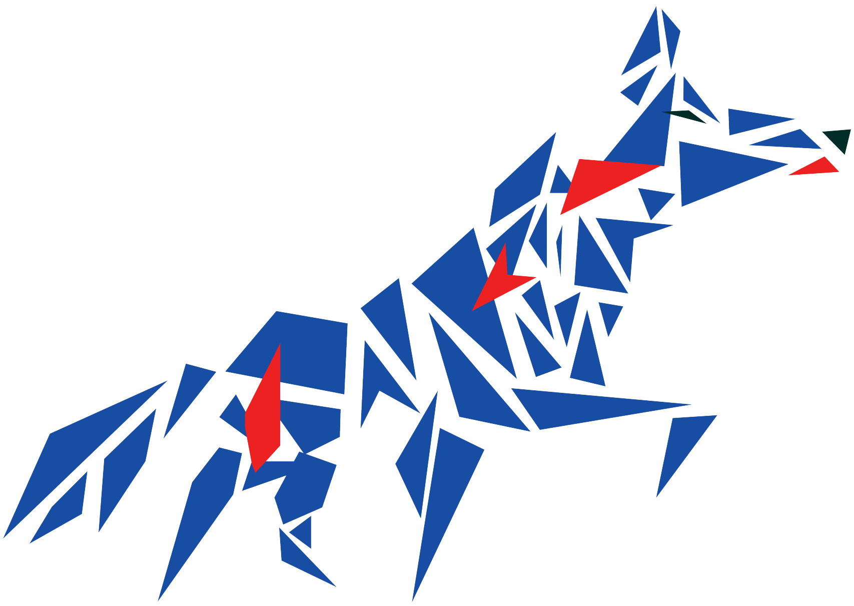
Innovative Smart System

Portefolio

Clément Gauché



Institut National des Sciences Appliquées de Toulouse

January 13, 2025

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Contents

Chapter I.

Introduction

Ce portfolio est organisé de la façon suivante.
Chaque section correspond a une matière, et est décomposée en sous parties:

- Context
- Technical summary
- Pratical Work
- Competences gained
- Analysis and remarks
- Reflections

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	Trainee in the network department, Implementation of monitoring functions in a Software Defined Network (SDN)
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. Cloud and Edge Computing

2.1 Context

Among the first few 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 ubiquitous computing. By nature, the course had both a theoretical and hands-on combination that gave an insight into how such paradigms shape the current IT systems and their applications.

2.2 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.

Another important area discussed in this course was cloud computing: a discussion of definitions, among them the widely recognized framework from NIST, but also established its core characteristics. On-demand service provision, scalability, pay-per-use, and supply of services in models like IaaS, PaaS, and SaaS-all these form key elements that make cloud computing. We also looked at edge computing, which is integral to the IoT ecosystem, and allows for latency reduction and ensures data sovereignty. However, this also presents challenges of connectivity and the processing power of the edge devices. Finally, we discussed fog computing: an intermediary layer that connects cloud services with data sources, reducing the distance data needs to travel, and allowing real-time interaction. This was aptly referred to as the process of "cloudifying the IoT world."

2.3 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. These have provided a 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.

2.4 Competences gained

The course really enhanced my technical capability and comprehension of Distributed Systems. Cloud and Edge computing architectures focusing on high performance and dependability. My experience with VirtualBox and OpenStack gave me hands-on experience in creating and managing virtual environments: virtual machines and containers. This also includes the development of expertise in deploying and orchestrating containerized services through the use of platforms like Kubernetes, which is critical in modern-day designing of distributed systems. The course enhanced my system design skills for low latency and high resilience, particularly in IoT and Distributed Environments. Apart from that, I learned real-time disruption management at the edge and continuity of service, despite variability in edge nodes.

2.5 Analysis and remarks

One of the strong points of this course is the structuring of the practical sessions, which did combine theory and practice quite nicely. Each session started with a theoretical introduction to reinforce the central concepts of the course; we then had to go through an implementation phase by putting these ideas into practice with tools such as VirtualBox and OpenStack APIs. In this way, both my understanding was solidly shaped and practical skills were gained; hence, learning has been both engaging and very effective. I found this format particularly enriching because it mirrors real-world workflows, where a deep understanding of concepts must complement practical expertise.

2.6 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 designing and managing distributed systems in professional life. In this learning process, I developed not only my technical skills but also the problem-solving approach needed to handle modern IT system complexities.

3. 5G technologies

3.1 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.

3.2 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.

3.3 Skills acquired

3.4 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.

4. Service Architecture - Software Engineering

4.1 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.

4.2 Technical

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.

4.3 Practical work = Projets de TP

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, handling data formats (e.g., JSON, XML), and exploring HATEOAS.
- 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.

4.4 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:

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..1: Competences Gained During the Service-Oriented Architecture Course

4.5 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. Even though this was demanding, the tutorials were essential in building a strong technical foundation. They covered key concepts like service discovery, load balancing, and HATEOAS, which were crucial for understanding 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.

5. Wireless Sensor Networks

5.1 Context

Wireless sensor networks (WSNs) are to me an essential part of the development of intelligent, interconnected systems such as the Internet of Things (IoT). These networks consist of spatially distributed sensor nodes that monitor physical or environmental conditions and relay the data wirelessly for analysis. Unlike traditional networks, WSNs prioritize energy efficiency, robustness, and scalability due to the constrained nature of their hardware and the challenging environments in which they often operate.

The course on WSNs, taught by the Professor Daniela Dragomirescu, 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 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 (5G), and M2M (5G) protocols.

5.2 Technical

In this course I discovered the unique specifications of WSN protocols such as Zigbee and Bluetooth, 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 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.

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.

5.3 Practical work

5.4 Skills acquired

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

- **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).
- **Simulation and Evaluation:** Using simulation tools to model and assess network performance under diverse environmental and operational constraints.
- **Energy Optimization:** Developing strategies to balance energy efficiency and communication reliability in WSNs.
- **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).
- **Problem Solving in Constrained Environments:** Designing practical solutions for challenging real-world scenarios, such as our sewer pipe monitoring project.

5.5 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 our project also helped me understand the practical challenges of deploying WSNs in constrained environments. For our sewer pipe monitoring system, we needed to balance low energy consumption, reliability, and scalability. While designing the 3N-MAC protocol, I gained hands-on experience in tailoring a solution to meet the specific demands of the application. However, one major difficulty was the implementation of the physical layer using GNU Radio. Although we had a brief introduction to the software, it was not sufficient to fully understand how to implement a working physical layer. For future projects, I think it would be extremely useful to have more detailed guidance or tutorials on how to use tools like GNU Radio to implement protocols effectively.

Another takeaway from this course is that there is no one-size-fits-all solution for WSNs. For instance, energy-efficient protocols like B-MAC are ideal for low-traffic applications but can introduce delays or lack features like synchronization. Similarly, location-based protocols like GAF are great for dynamic networks but require accurate localization, which can complicate the 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.

6. Security for connected objects

6.1 Context

6.2 Technical

6.3 Practical work

6.4 Skills acquired

6.5 Analysis and remarks

7. Energy for connected objects

7.1 Context

The "Energies for Connected Objects" course dealt with pioneering strategies to feed IoT devices without batteries or wired systems. This training focused on Ambient Energy Harvesting and Wireless Power Transfer applied to the high demand for energy-autonomous solutions in healthcare, smart cities, and environmental monitoring, among others. It brought together a multidisciplinary approach, from theoretical insights to efficient, sustainable, and reliable energy system design.

7.2 Technical

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.

7.3 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.

7.4 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.

7.5 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.

7.6 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.

8. LAB at AIME

9. Microcontrollers Open-Source Hardware and Sensor Introduction

10. Communication protocols for LP-WPAN

10.1 Context

The LP-WPAN module, part of the IoT network architecture course, provided an in-depth understanding of low-power, short-range wireless communication technologies tailored for constrained IoT devices. The focus was on IEEE 802.15.4, which serves as a foundation for many IoT protocols like Zigbee, 6LoWPAN, and Thread. This standard facilitates the communication of devices with limited resources, such as sensors and actuators in smart homes, industrial monitoring, and environmental applications.

Throughout this course, I explored the intricacies of LP-WPAN, including its architecture, challenges, and specific adaptations for IoT networks. The emphasis on constrained environments—characterized by low bandwidth, high energy efficiency, and dynamic topologies—highlighted the unique challenges of enabling reliable and scalable IoT communications.

This module provided me with a solid theoretical foundation and hands-on experience in designing and evaluating LP-WPAN technologies, especially in the context of IPv6 adaptation.

10.2 Technical

The technical aspect of this module revolved around the design, characteristics, and operation of LP-WPAN networks and their interplay with the IPv6 protocol. Several key concepts were covered:

IoT Network Characteristics and Specificities

LP-WPANs are constrained by low power, small frame sizes (127 bytes max), and low data rates (up to 250 kbps). Networks are scalable, supporting dynamic addition and removal of devices, and rely on protocols like IEEE 802.15.4 for physical and MAC layers. Energy efficiency is a key focus, with devices often operating in low-power or sleep modes.

Rationale for Adopting IPv6 IPv6 Basics IPv6 Adaptation for IoT Networks IETF IPv6-Based Stack for IoT Existing IPv6-Based Network Technologies for IoT Relevance of IPv6 in the Semester Project IoT and Sustainability

Guided Tutorials

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.

10.3 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..2: Competences Gained During the LP-WPAN Module

10.4 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.

11. Embedded IA for IOT

12. Emerging Networks

13. Sensors Introduction

13.1 Context

13.2 Technical

13.3 Practical work

13.4 Skills acquired

13.5 Analysis and remarks

14. Master REOC

Chapter III.

Conclusion