CESI_
ÉCOLE D'INGÉNIEURS

DELIVERA BLE 1
PDCA 1 (Loop 1)

- **Title: SMART OCCUPATION** - *"Improve the efficiency of the usage of the coworking space"*
- **Team:** Joël-Stéphane YANKAM NGUEGUIM, Johan RINALDI, Matteo HEIDELBERGER, Kellian BECHTEL & Enorian RAJOELISOA
- **Group:** n°3
- **Area studied:** Co-working
- **Date:** 20/02/2026
- **Educational advisor:** DING Yuehua
- **Version:** v1.0.0

0) Executive summary

- **Observable problem**: Students have no way to tell if the co-working space is busy or not. It is often overcrowded or underused.
- **Functional requirements (SMART)**: Fast updates on occupancy rates, accurate data, secure data handling, low power consumption, safe to use at CESI.
- **Key constraints (latency, energy, network, security)**: Low latency, reliable transmission, frequent updates, low power consumption, encrypted & signed data transmission, reliable means of detecting/estimating occupancy. No fire risk.
- **Preliminary architecture choice**: We want to process data and determine the number of people for a specific room in real-time. So, as explained in section 4, we chose the **edge architecture**.

1) Observation & observable problem

- **Context/area:** The coworking space is a multifunctional environment used for lunch, work, conferences, and events. Depending on the time of day, occupancy levels vary significantly. During peak hours, such as lunchtime, some rooms become overcrowded while others remain underused. This leads to increased noise, reduced comfort, and lower productivity.
  Our IoT system aims to optimize room usage by providing real-time availability information.

Figure 1 - Target Area

- **Measurable facts:** Number of people per room | Number of rooms available | Average occupancy rate | Maximum capacity per room | Peak hours | Average duration of occupancy
- **Problem statement:** *"How can we optimize room occupancy in a coworking space by providing real-time availability information?"*
- **Beneficiaries/personas:**
  - Student
  - Employee
  - Visitors
- **Success indicators (KPIs):**

KPI	Description	Formula
Availability (%)	Percentage of time the system is operational and accessible.	$\frac{Total\ scheduled\ time}{Total\ operating\ time} \times 100$
Latency (ms)	Time required for occupancy data to be transmitted and displayed.	$Time_{received} - Time_{sent}$
Room utilization (%)	Percentage of time a room is occupied during its availability period.	$\frac{Total\ available\ time}{Occupied\ time} \times 100$
Energy consumption (kWh)	Total energy used by IoT sensors over a period of time.	$Power\ (kW) \times Time\ (h)$

2) Functional requirements (SMART)

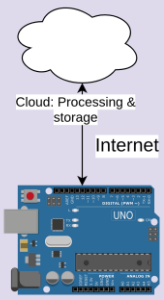
- The system shall detect room occupancy using at least one motion sensor per monitored room.
- The system shall transmit occupancy status updates at least every 60 seconds.
- The system shall process occupancy data locally before transmission to ensure privacy and reduce unnecessary network traffic.

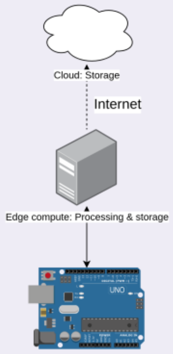
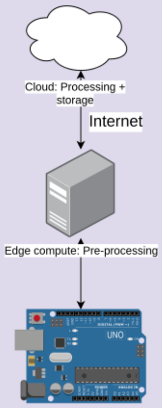
- The system shall securely transmit data using encrypted communication protocols.
- The system shall display real-time occupancy status on a cloud-based dashboard accessible to users.
- The system shall operate automatically without requiring user interaction.

3) Constraints & assumptions

- **Latency/local criticality**: The system must provide occupancy updates with a maximum delay of 2 seconds to ensure near real-time information for students outside the coworking space. It must remain operational even in case of temporary network failure; therefore, data processing and decision-making are handled locally using an edge computing approach.
- **Energy/autonomy**: Devices must operate autonomously for at least one working day, with a target autonomy of several weeks. To reduce energy consumption, devices operate only during opening hours (8:00 AM - 6:00 PM), use low-power communication protocols, and transmit data only when occupancy status changes.
  Due to budget constraints in this initial design phase, battery-powered devices are selected to reduce installation costs and simplify deployment.
- **Network/coverage**: The system must cover the entire coworking space and remain accessible.
- **Criteria**: Ensure electrical and operational safety in compliance with building regulations.
- **Interoperability**: The system supports open communication protocols such as MQTT or CoAP to ensure interoperability and future scalability.
- **Security (baseline)**: The system follows baseline IoT security principles inspired by ETSI EN 303 645. No personal data is stored. Occupancy measurement is anonymous and processed locally to ensure privacy protection.
- **Assumptions (what we assume to be true at this stage)**: At this stage, we assume that students may attempt to tamper with the system, the campus Wi-Fi is stable, devices operate on battery power due to budget constraints, the project budget is limited, and the IT department approves network integration.

4) Architecture benchmarks (Cloud / Edge / Hybrid)

Option	Quick diagram	Latency	Simplicity	Limitations	Go/No-Go
Cloud		High: all operations are delayed by the latency to the cloud	Simple: Existing cloud platforms can be leveraged	High latency, more bandwidth required as all information is sent to the cloud, potential offline limitations if connectivity is interrupted, as well as regulatory issues concerning data locality.	Bad if the embedded device needs to wait for processing done on the cloud.

Edge		Low: The decisions are made close to the embedded devices	Medium complexity: the edge compute needs to be installed and set up	Less processing power, cross-site data aggregation is harder, resource constraints in term of compute power or storage	The deployment of edge compute needs to be possible If data from multiple sites needs to be aggregated, this approach is disadvantageous
Hybrid		Medium: some decisions can be taken directly on the edge, but others still experience the latency of the cloud. Less data needs to be transmitted to the cloud as pre-aggregation and filtering can be done on the edge.	High complexity: the edge compute. needs to be installed and set up. Responsibilities need to be split between the edge and cloud, and an additional layer (fog) is needed.	The complexity leads to higher operational overhead.	The deployment of edge compute needs to be possible This approach is only useful if the data can be pre-processed and/or if decisions can be taken on the edge

Analysis & provisional choice: We chose an edge computing architecture because it ensures low latency and real-time occupancy detection. Data is processed locally by the microcontroller connected to each motion sensor, reducing network dependency, and improving system reliability.

5) Techno-communication comparison (LTN/LPWAN vs. short range)

Family	Examples	Range	Data rate	Consumption	Cost/infrastructure	Remarks - use cases
LPWAN	LoRaWAN, Sigfox, NB-IoT, LTE	2–15 km urban, up to ~40 km rural (depending on tech & environment)	Very low (≈ 0.1–100 kbps)	Ultra-low (battery life 5–15 years typical)	Low device cost, infrastructure depends (private gateways or operator network)	Designed for IoT sensors sending small, infrequent data. Smart meters, agriculture, asset tracking, environmental monitoring.
Short range	Bluetooth Low Energy (BLE), Zigbee,	Centimeters to ~100 m (someti	Low to high (≈ 100 kbps to hundreds	Low to high (BLE very low, Wi-Fi higher)	Minimal infrastructure (often just a gateway or access point)	Best for local connectivity. Smart home devices, wearables, industrial

	Wi-Fi, HaLow, NFC	mes a few hundred meters line-of-sight)	of Mbps depending on tech)			automation, consumer electronics.

Protocols overview

PROTOCOL	CATEGORY	SCOPE	FLOW RATE	CONSUMPTION	MOBILITY	NETWORK TYPE
Zigbee	Short range	Indoor/building	250 kbps	Very low	Low	Mesh
BLE	Short range	Indoor/personal area	2 Mbps	Very low	Low	Star/Mesh
LoRaWAN	Long range	Outdoor/urban & rural	0.3-50 kbps	Very low	Low	Star-of-stars
Sigfox	Long range	Outdoor/urban & rural	100 bps	Ultra-low	Low	Star
Wi-Fi	Short range	Indoor/local area	11 Mbps-9.6 Gbps	High	Low-medium	Star (AP)
LTE-M	Long range	Cellular-wide coverage	1 Mbps	Low-medium	High	Cellular star
NB-IoT	Long range	Cellular-wide coverage	250 kbps	Very low	Medium	Cellular star

Multi-criteria conclusion: Because of our project context and the limited size of the coworking space, a short-range communication protocol is the most appropriate solution. Zigbee is particularly relevant due to its low energy consumption, reliable mesh networking capability, and sufficient range for indoor environments.

Zigbee overview

CRITERION	VALUE
Flow rate	Up to 250 kbps
Range	10-100 m indoors

Frequency	2.4 GHz globally, 868/915 MHz in some regions
Topology	Star, Tree, Mesh
Latency	30-50ms
Power consumption	Very low (months/years on battery)
Security	AES-128 encryption, authentication
Ownership	Open standard

6) Initial risks & measures

- **Technical:** In our IoT system, connection loss could prevent occupancy data from being transmitted to users or the cloud, insufficient network bandwidth could delay updates, and low availability of sensors or microcontrollers could interrupt data collection.
- **Safety**: In this project, data leaks could expose room for occupancy information, potentially violating privacy. Additionally, electrical faults in sensors or wiring could pose safety hazards.
- **Operational**: Theft or vandalism of sensors could disrupt operation, and misuse by users could compromise data accuracy.

7) Summary & preliminary selection

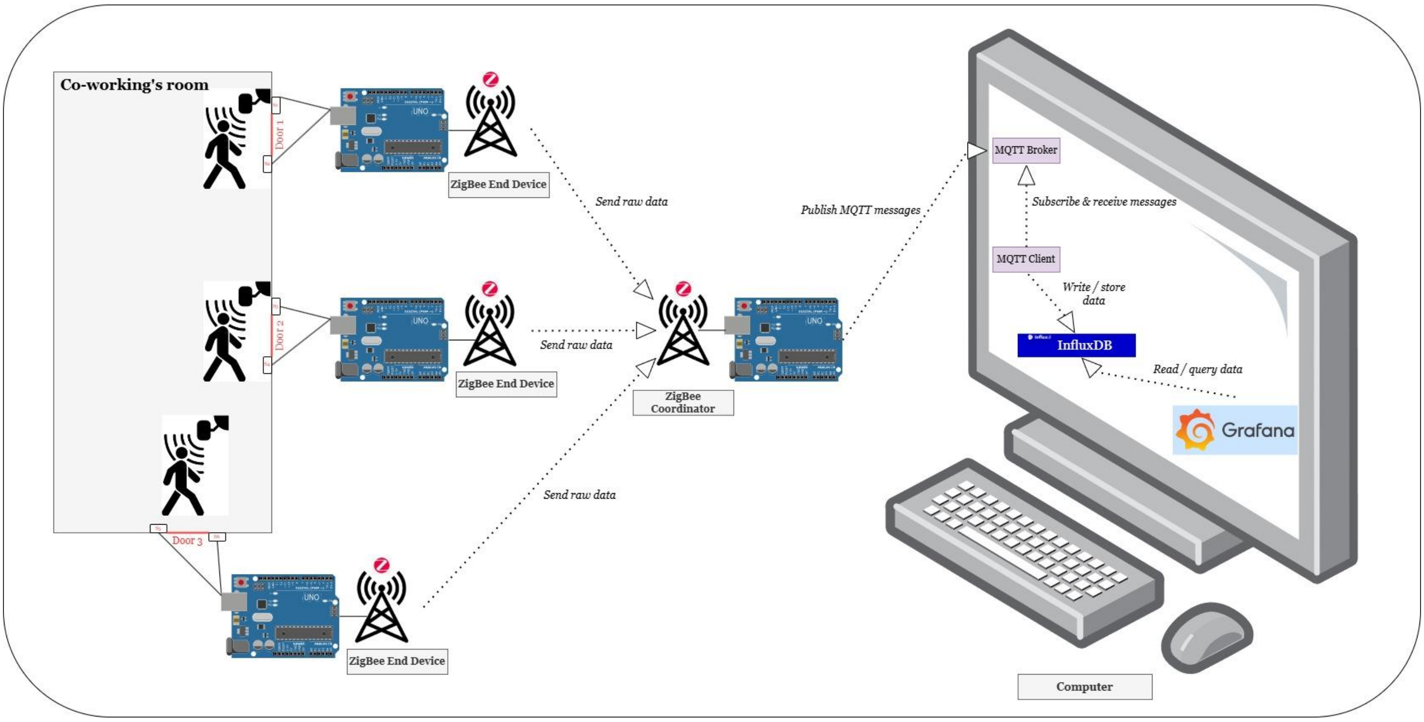
- **Selected architecture** (provisional): The provisional architecture consists of one or two motion sensors per door; each connected to a local Arduino. These "door" Arduinos send occupancy data via Zigbee to an **Arduino coordinator**, which acts as an edge processor. The coordinator converts the Zigbee frames into **MQTT messages** and publishes them to a broker hosted on a PC. The broker then stores the data in a database (example: InfluxDB), which is read by Grafana to visualize real-time room occupancy.
- **Prospective technology family**:
  - ➢ **Sensors:** PIR motion sensors for occupancy detection.
  - ➢ **Edge Processing:** Arduino microcontrollers per door, with one Arduino acting as Zigbee coordinator.
  - ➢ **Network Protocol:** Zigbee mesh network for local communication.
  - ➢ **Communication Format:** MQTT messages for broker communication.
  - ➢ **Storage & Dashboard:** Database (InfluxDB) and Grafana dashboard for real-time monitoring.
- **Minimum security requirements**:

  Zigbee communication:
  - ➢ AES-128 encryption with MIC (Message Integrity Code)
  - ➢ Frame counter checks to prevent replay attacks
  - ➢ Trust Center setup, with Permit-Join disabled after installation
  - ➢ ACL (whitelist) and periodic network key rotation

  MQTT / Broker:
  - ➢ TLS-secured communication
  - ➢ Authentication for broker access

➢ Secure storage of encryption keys

- **Schematic diagram** (embedded image or link):



Schema 1 - System Architecture Overview

8) References

- LoRa Alliance (2015), LoRaWAN specification, performance, and deployment guides

- Sigfox (2026), Technology overview, coverage, power profile

- Bluetooth SIG (2026), Bluetooth Low Energy core specs and range/power characteristics

- Wi-Fi Alliance (2026), Wi-Fi families (including HaLow / 802.11ah)

- Connectivity Standards Alliance(2026), Zigbee specifications and use cases

- Gomez, C., Oller, J., Paradells, J. (2012). Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology. Sensors.

- TBO Editorial (2024). IoT Data Processing Architectures: A Beginner's Guide to Edge, Fog & Cloud Pipelines

- Sami TABBANE, ITU-T, (2018), IoT Standards, Part I: IoT Technology and Architecture