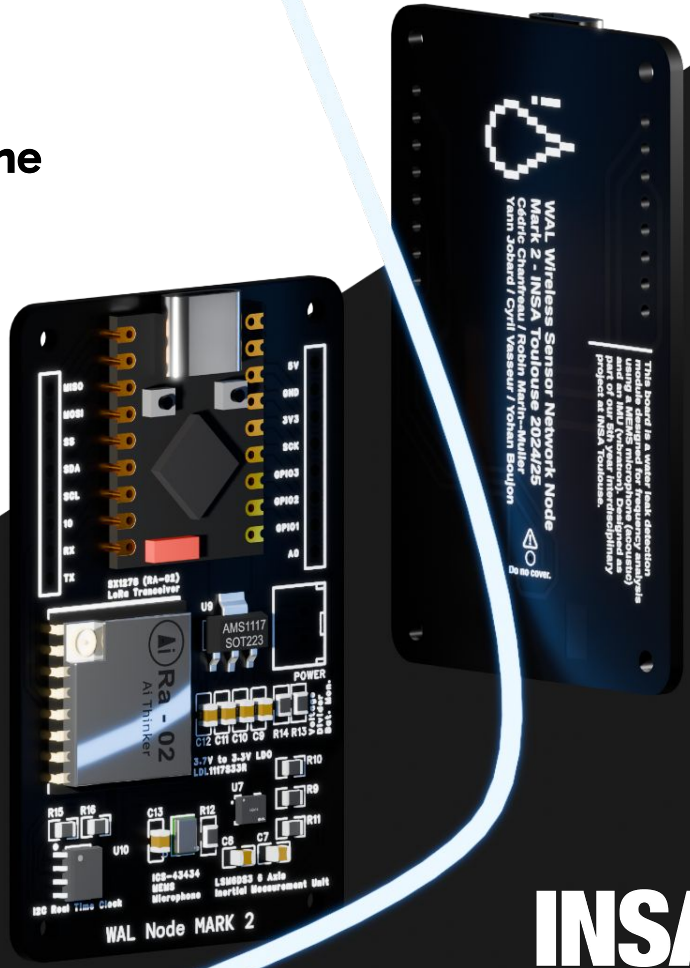
23 Jan 2025

# Non-Intrusive Detection of Water Leaks in Pipeline Networks Using Vibration and Acoustic Sensors

*An Innovative Project*

PTP Innovative Smart Systems





Yohan Boujon
Robin Marin--Muller
Cyril Vasseur
Cédric Chanfreau
Yann Jobard

*Supervised by William* Perez


INSA
TOULOUSE

# Summary

1 **Water Leak Detection System**

Detect and locate leaks in a non-intrusive way.

2 **Sensors, Signal Analysis and Hardware**

Collecting data, analysing signals and predicting leaks.

3 **Node-to-Node Communication**

Communication protocol, securing the data, custom packets.

4 **Processing Data on the Server**

Subscribe-publish data, storing it in a database.

5 **Displaying the Informations**

User interfaces: Web and Mobile applications.

6 **Project Results and Limits**

Power efficiency, Achievements and Future Work.

**What A Leak**

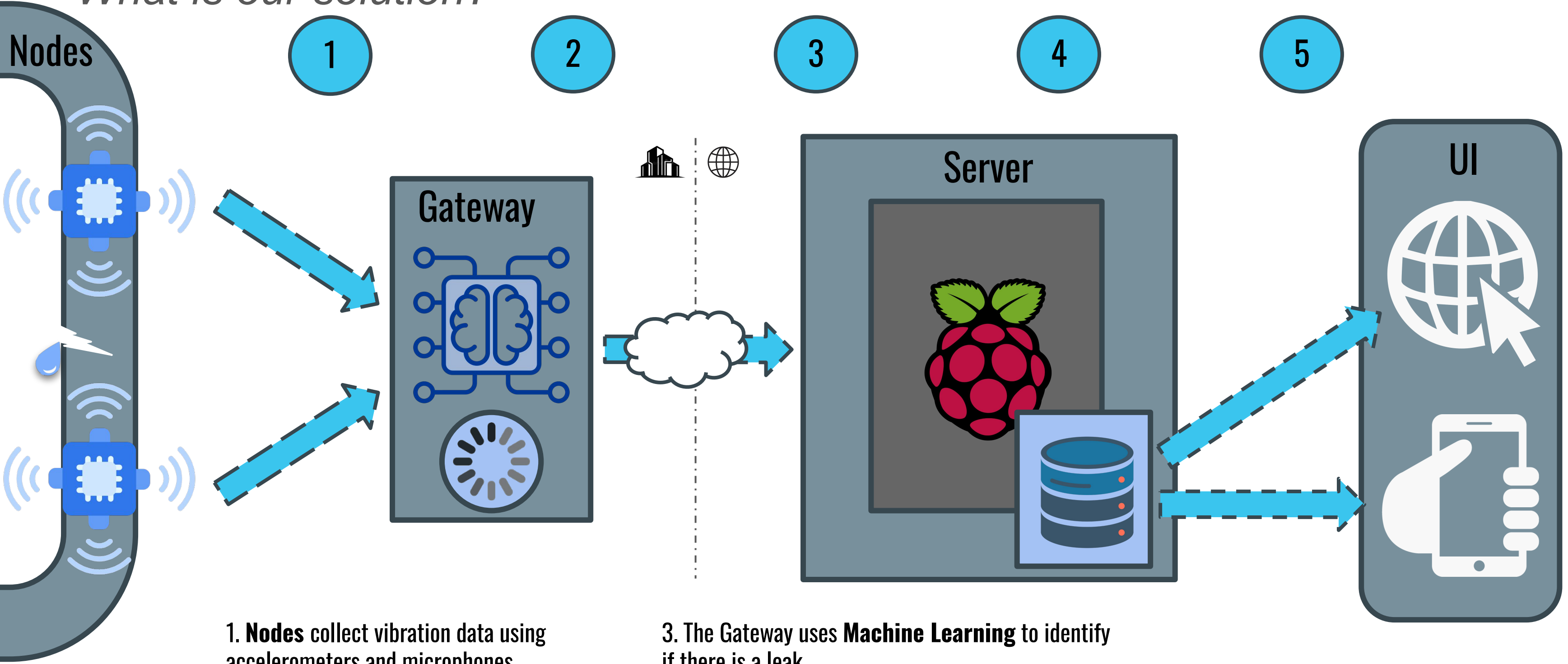
# Problem Statement

- In France, **20%** of the water transported through the distribution network is lost due to leaks. (**1 billion m³**)
- For public infrastructures leaks cost more than **4 billion euros in France.**
- Traditional methods are costly and inefficient.


Yann

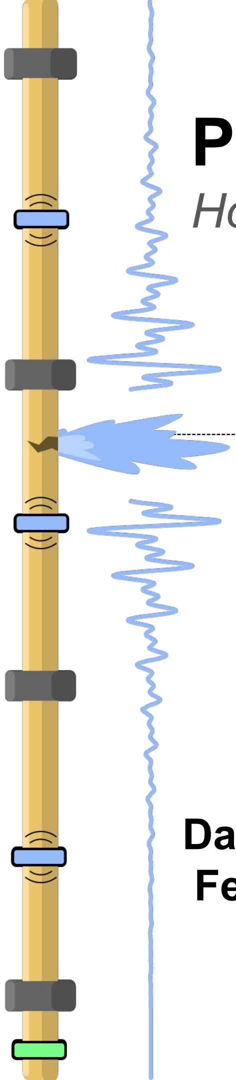
# Overview

*What is our solution?*




1. **Nodes** collect vibration data using accelerometers and microphones.
2. Data is sent via LoRa to the **Gateway.**
3. The Gateway uses **Machine Learning** to identify if there is a leak.
4. The data is sent to the Server through the web to be stored in the **Database.**
5. **User Interfaces** (Website and App) retrieve relevant information and conclusions from the database.

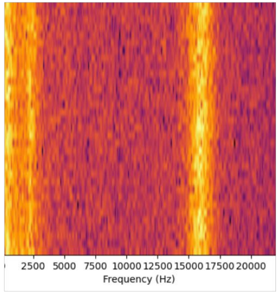
Yann

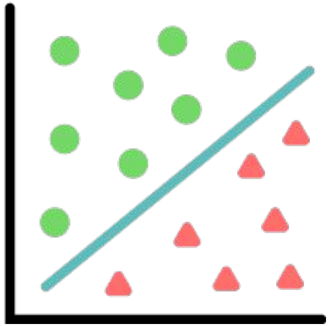


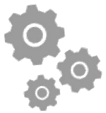
# Principle and Architecture

*How can we identify leaks, what principles can we use?*

Leaks emit distinct spectral signatures





⚠ Leak

✓ No Leak

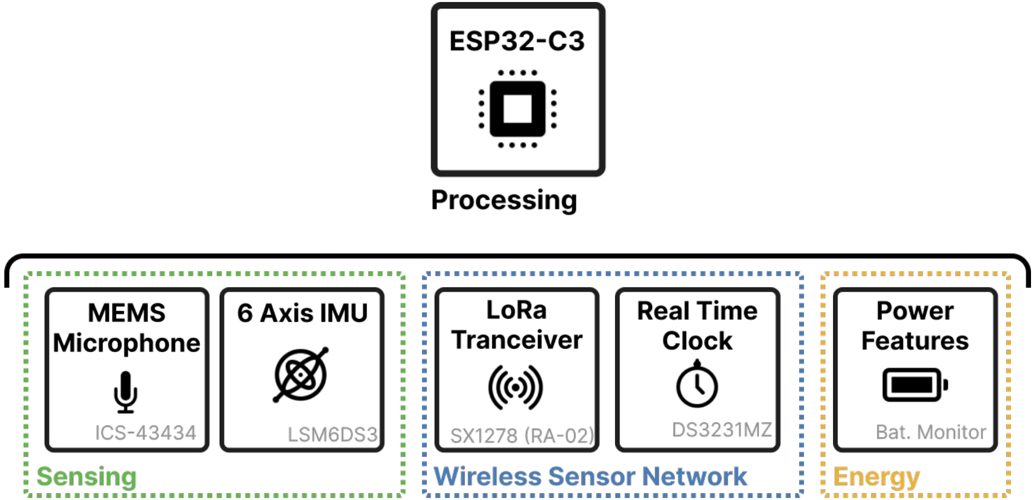
**Data Collection and Feature Extraction**

**SVN Machine Learning Classifier**

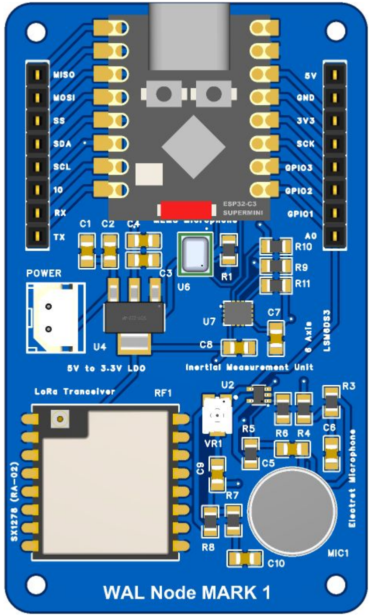
**Model Prediction**

# Hardware Architecture

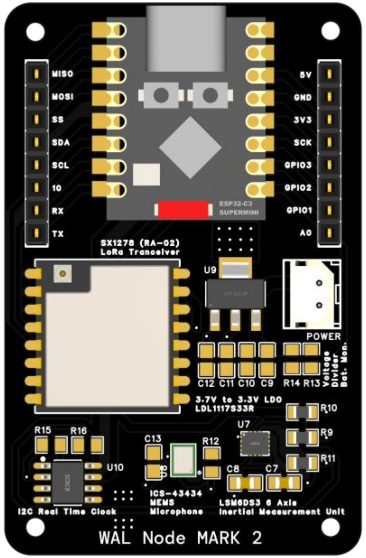
*Which hardware is required, for effective leak detection and WSN integration?*




**Hardware Architecture**




**PCB Rev 1**




**PCB Rev 2**

# Machine Learning

*How to improve leak detection accuracy?*

**Traditional Methods?:** Like Threshold-Based Detection: Unsuitable and hard to implement due to unpredictable noise.

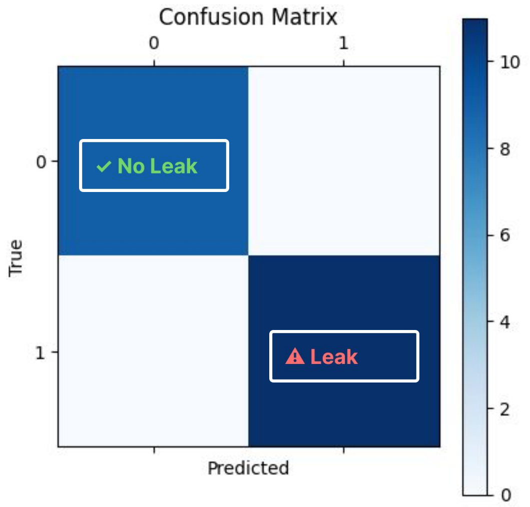
---

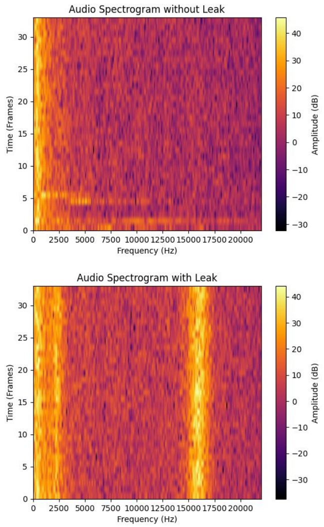
***Machine Learning Approach***

**Feature Extraction:** Fast Fourier Transform (FFT) to identify frequency patterns unique to leaks.

**Algorithm:** Support Vector Machines (SVM) for classification.

**Good Training Data = Good Results**




**ML Results**

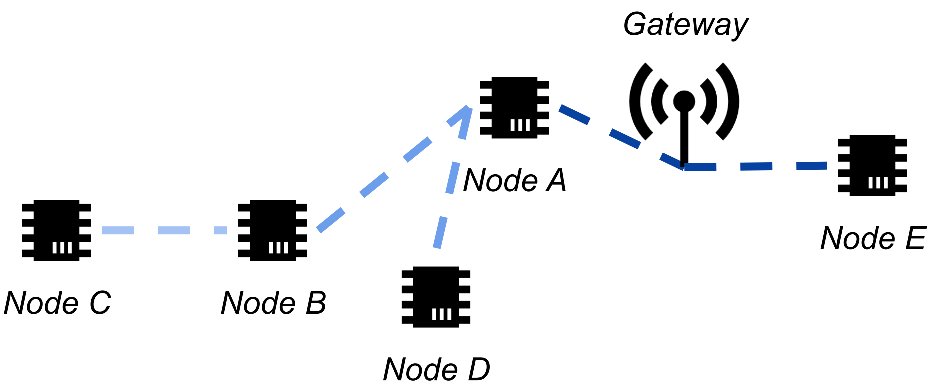
# LoRa Communication

*Which method is best for communication?*

- **Choosing the Right Frequency**:
  - LoRa on *433MHz* can go through many different materials.
  - *(Possible Upgrade) Lower frequencies* with a custom implementation.

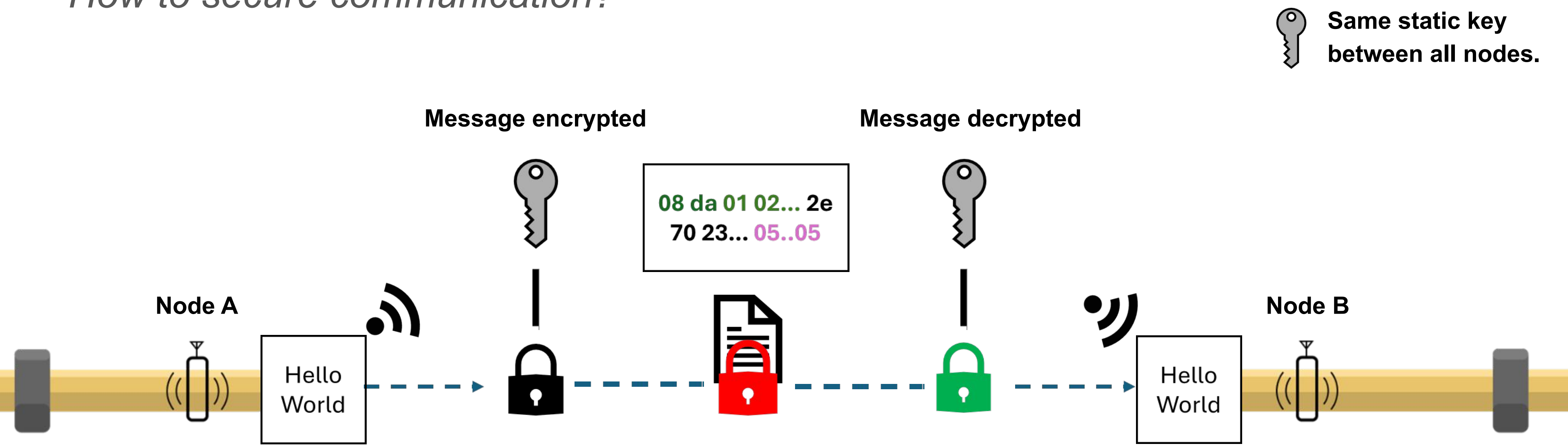
- **266m** in an unobstructed area
- **35m** with 2 well-isolated floors

- **Mesh Network Structure**:
  - Each node can communicate with its *neighbor.*
  - *Sending* its information until the gateway is reached.



# Cryptographic Security

*How to secure communication?*




**IV (Initialisation Vector):**

- Minimal Overhead (2 Dynamic 14 Static Bytes).
- Add Randomness to Encryption (Prevents Replay Attacks).
- Unique Encrypted Data.
- Enhanced Data Security.

**Padding :**

- Extra Bytes to Data.
- Block Size Alignment (16 Bytes).

# Custom Protocol

*How can we exchange data?*

- **Using the less possible memory**:
  - 256 Bytes *maximum* can be send each time.
  - It takes *multiple milliseconds* to send this data each time.


What A Leak Protocol:
A Resilient MAC Protocol for Efficient Communication in Harsh Environment
Rev 1.0

WAL Protocol

*before topology* — *10 bytes*

| WAL Header *(3 bytes)* | Address ***SRC*** *(3 bytes)* | Address ***DST*** *(3 bytes)* | Command *(1 byte)* |
|---|---|---|---|

*after topology* — *6 bytes*

| WAL Header *(3 bytes)* | Node ***SRC*** *(1 byte)* | Node ***DST*** *(1 byte)* | Command *(1 byte)* |
|---|---|---|---|

*The destination changes at every node.*

- Topology Payload
- Time Payload
- Data Payload
- Battery Payload

# Custom Protocol

*How can we exchange data?*

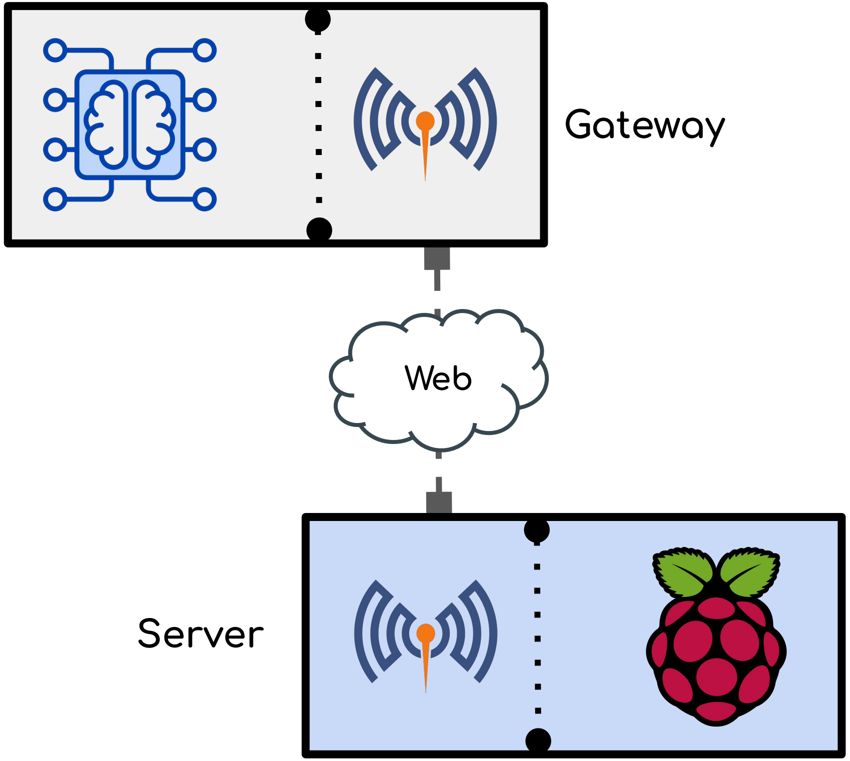
- **Interpreting the data**:
    - The data has to be *easily* readable for each operation.
    - *MQTT* with a simple publisher can transfer the data to the *internet.*

# Server-Side Processing

*How does the data reach the server?*

- **Data Reception through web**:
  - MQTT Client Gateway.
  - MQTT broker receives sensor data.

- **Security:**
  - MQTT communication limited by credentials.
  - Firewall configuration.
  - On reception, server throws inadequate Data received.



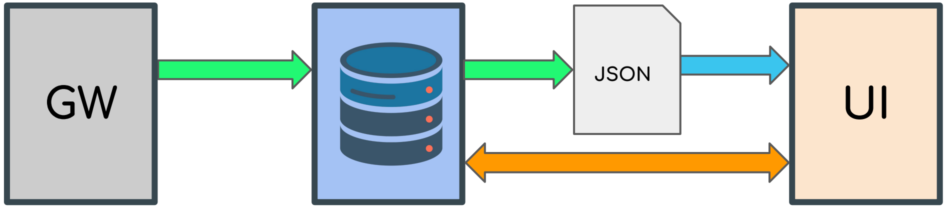
# Server-Side Processing

*How de we store the data?*

- Dynamic Data
  - **CHANGING OFTEN**
  - Node ’s ID
  - Measures:
    - **FFT**
    - **Status**
    - **Timestamp**
  - Accessory:
    - **Temperature**
    - **Battery**

- Static Data
  - **RARELY CHANGING**
  - Node ‘s ID
  - Location
  - UI Settings

- **Processing**:
  - Format static data into JSON for visualization on web and mobile platforms.



# Web interface

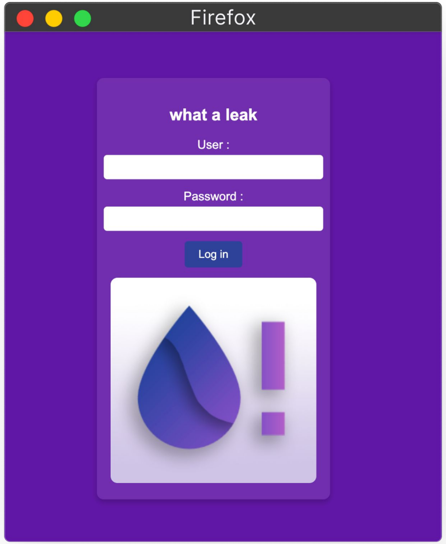
*How can we show the collected data to the user?*

**Web Application**:

- Developed with **Spring Boot** and **Maven.**
- **Accessible** from everywhere.

**Real time update:**

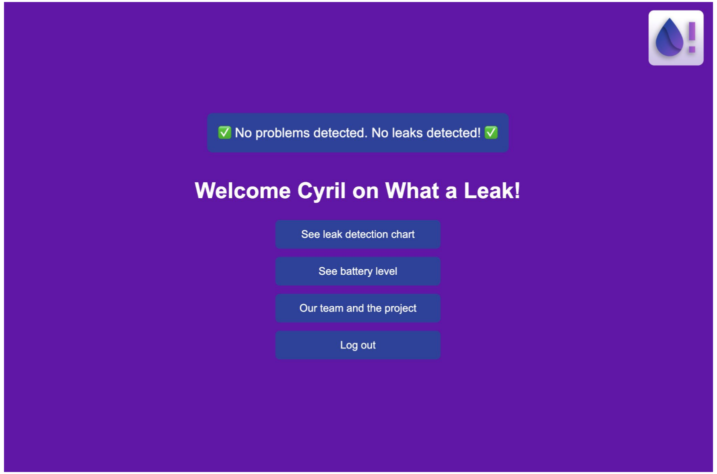
- Each update on the DB.




User Login Screen

# Web interface features

*How can we show the collected data to the user?*

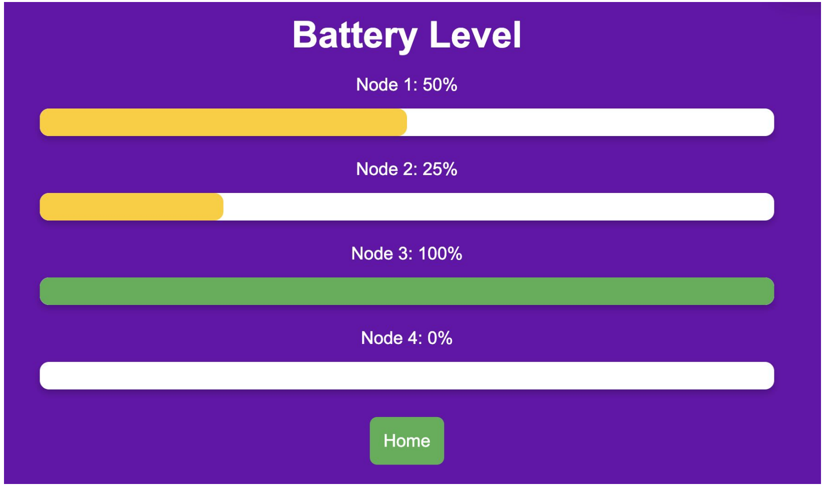



**Real Time Alerts**

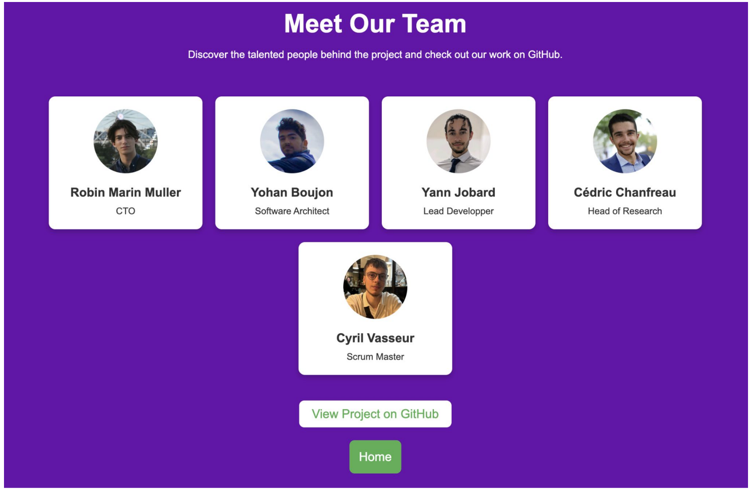

**FFT Visualisation**

# Web interface features

*How can we show the collected data to the user?*




**Battery Level Monitoring**




**The Team**

# Mobile application

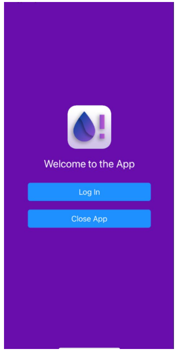
*How to make it more user-friendly?*

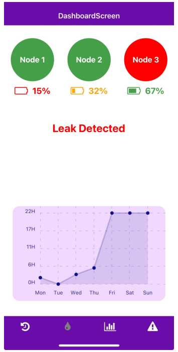
**Main Screens**:

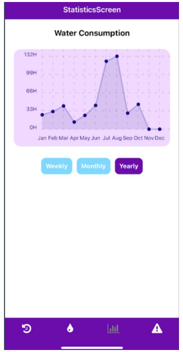
- **Home Screen:** User authentication (Log-in / Log-out).
- **Dashboard Screen:** Real-time node status (battery level…).
- **History Screen:** Node mapping over the network pipe.
- **Statistics Screen:** Water consumption over time.
- **Leak Screen:** Alerts and leak history.

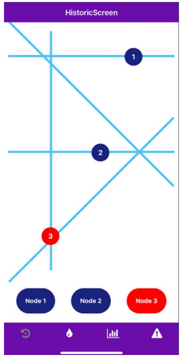
**Key Features**:

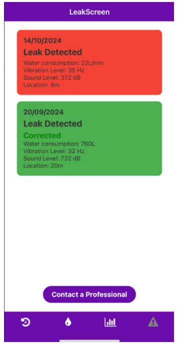
- **Cross-Platform**: **React Native & Expo** for Android and iOS.
- **Intuitive Interface**: User-friendly dashboard for real-time monitoring and quick actions.
- **Data Synchronization**: Integration with the server for live updates.
- **Comprehensive Tracking**: Leak history, node status, call a professional and water usage.



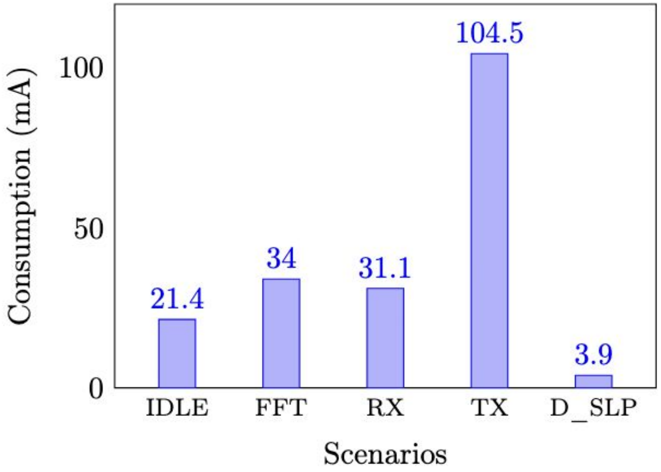



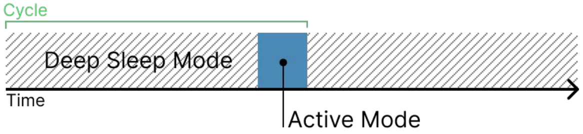




Cedric

# Energy Management

*What is the biggest limit of the system?*




**Energy Consumption of WAL Node V1**

**Real Application with two *INR18650-25R* batteries:**

- **Capacity of one battery**: $2500\,\mathrm{mAh}$
- **Two batteries in parallel**: $2500 \times 2 = 5000\,\mathrm{mAh}$
- **Node consumption**: $3.9\,\mathrm{mA}$

$$\frac{\text{Total Battery Capacity}}{\text{Node Consumption}} = \frac{5000}{3.9} \approx 1282\,\text{hours}$$

$$\boxed{\frac{1282}{24} \approx 52\,\text{days}}$$

Cyril

# Challenges and Future Work

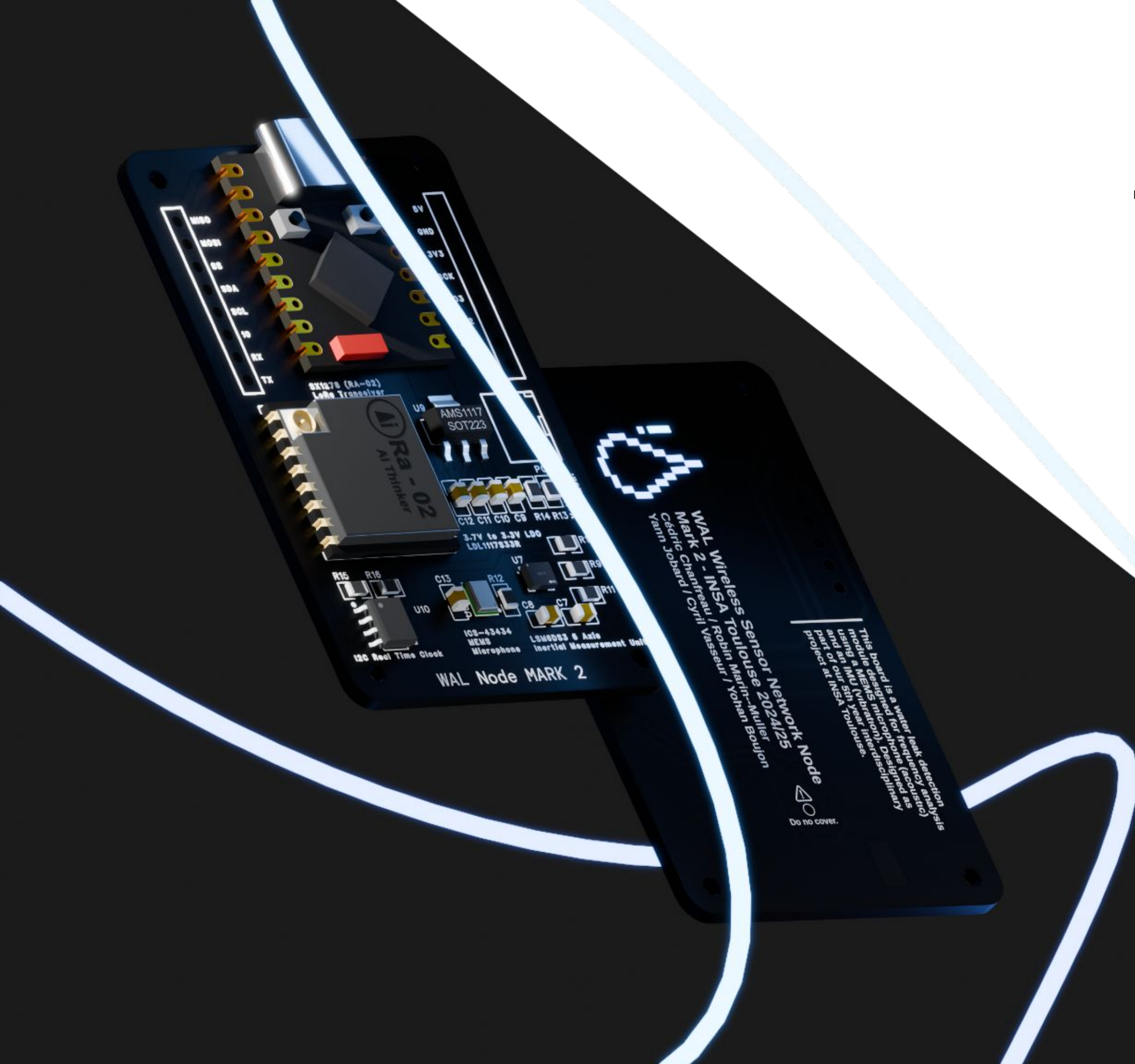
*What are all our current limits?*

| Challenge | | Future work |
|---|---|---|
| No battery management. | → | Inductive charging and optimised consumption. |
| Signal interferences in high-noise environments. | → | Refine AI models for better accuracy. |
| Untested prototype under real-world conditions. | → | Install the network in an expanded and more realistic prototype model. |
| Current physical model prototype not as effective has we wanted. | → | Redesign/build *(asking help to a professional).* |

# Results

*What have we accomplished?*

- Controlled tests confirmed accurate leak detection and signal classification.
- Successful LoRa communication even in noisy environments.
- Effective node to website/mobile application communication.
- Succeeded using the Agile method.
- Mark 1 fully working. Mark 2 in progress.
- Great teamwork.




**Thank you for listening!**

# Any Questions?

# Kick Starter !