Mobility platform in Aveiro Tech City Living Lab Infrastructure

Final Presentation

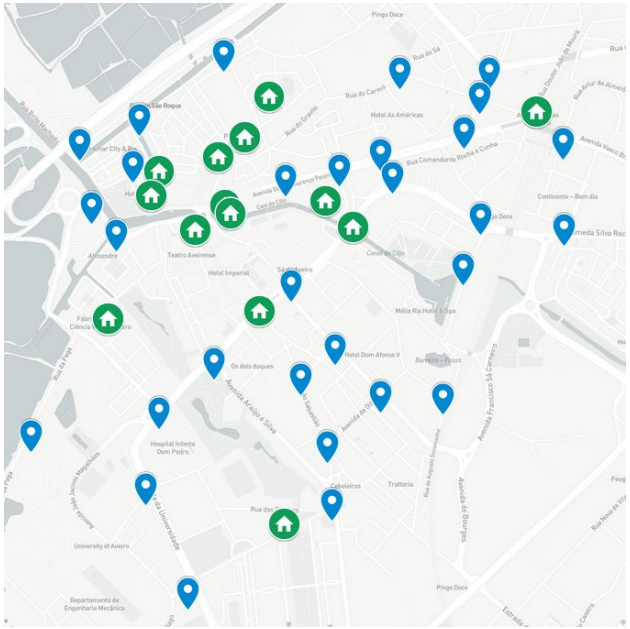
Diogo Mendes, 88801
Hugo Leal, 93059
Luísa Amaral, 93001
Maria Cunha, 93089
Pedro Loureiro, 92953

Supervisors:
Dr.ª Susana Sargento
Dr. Pedro Rito

Introduction

Context

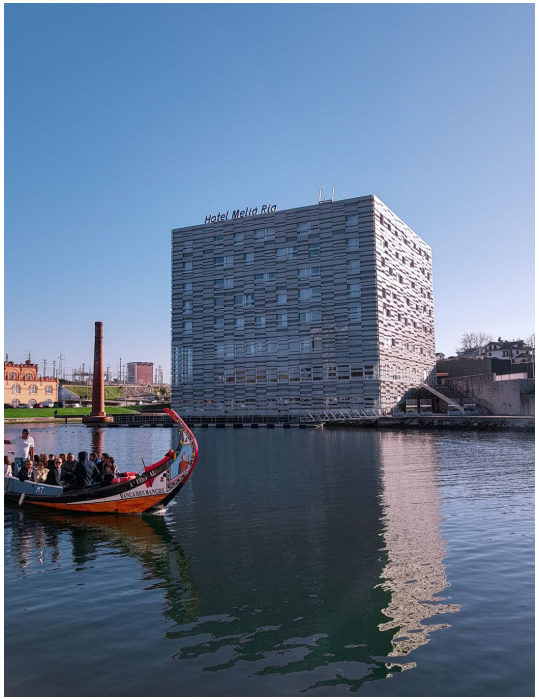
- In Aveiro there are 44 stations installed and interconnected with fibre
- The stations contain environmental sensors, radars, LIDARs, video cameras and computer edge units
- This infrastructure is connected to the data center in IT
- **Goal**: build a tech city living lab as a complete connected city to experiment future services and applications in a real environment



green home → buildings
blue pins → smart lamp posts

Problem

- Enrich the Aveiro Tech City Living Lab (ATCLL) infrastructure in order to improve the lifestyle of citizens and for research purposes
- The Aveiro's Town Hall proposed a task:
  - To monitor people in public spaces
  - To monitor the quantity and frequency of *moliceiros* in the *Ria de Aveiro*
  - To monitor people in the *moliceiros* in the *Ria de Aveiro*



Initial goals (I)

- Develop solutions to get values from the sensors and understand which type of sensors to use
- Develop mechanisms to detect people, *moliceiros*, movement, etc.
    - Use of different sensors
        - WiFi devices detection
        - Video AI for object detection
    - Sensor fusion and processing

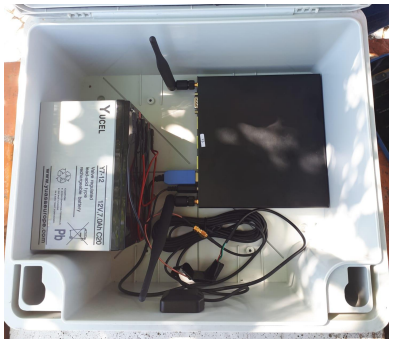
Initial goals (II)

- Develop a subscription broker to accumulate past and real time data
- Develop a web application to show the subscription broker’s data
- Overall data processing, persistence, analytics and visualization in the web application
- Improve team-working skills and learn how to work with new technologies and equipments
- Documentation writing

How to?

- Collect information regarding the number of people, cars, bicycles and moliceiros in certain areas of the city
- Use of object detection models to count the objects from the live video of cameras
- Use of Wi-Fi sniffing to count the number of devices near the smart lamp post
- Display of the data in a dashboard so that it may be observed by the public

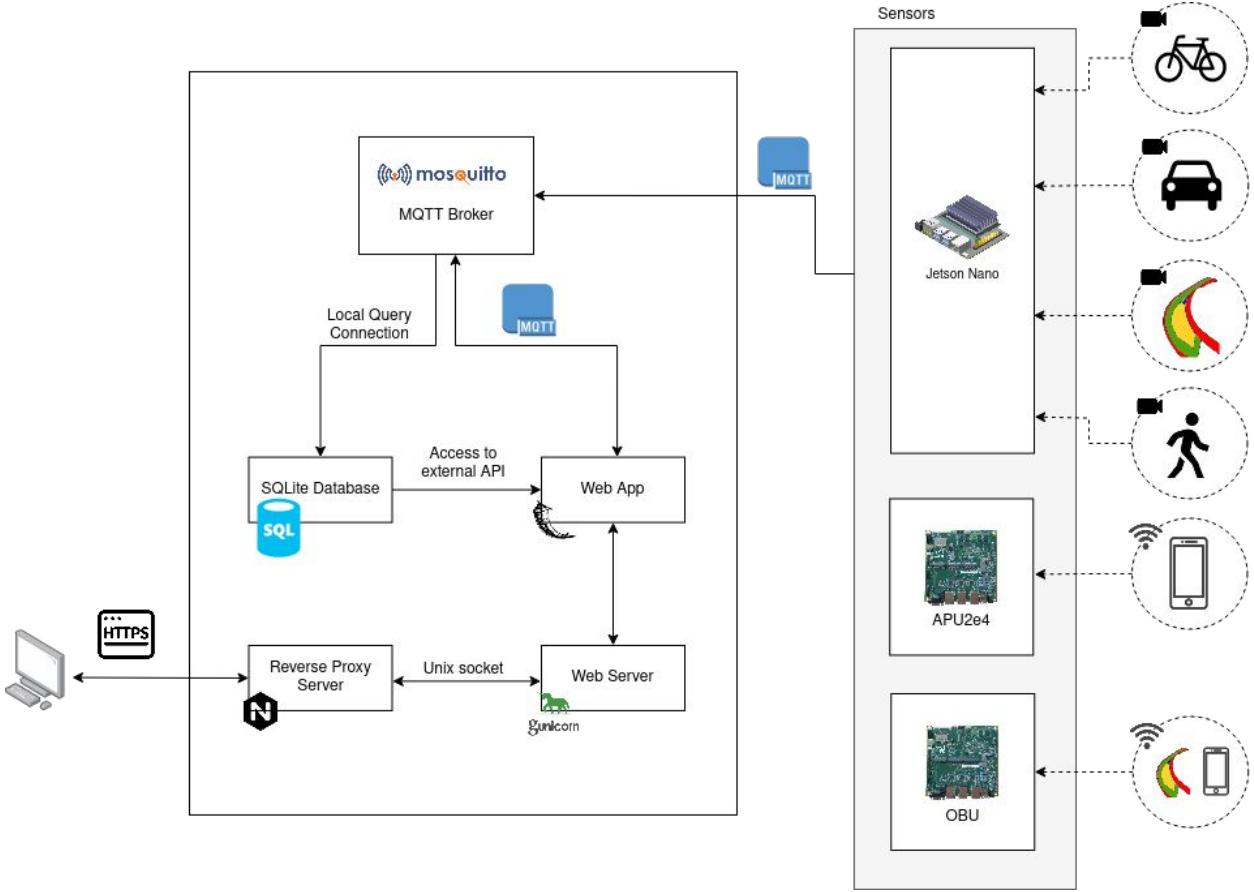
Our resources



Expected Results

- Detection mechanisms:
  - Vision and communication-based
  - Objects and movement
  - Aggregated detection
- A web application to present the results obtained in real-time as well as past results
- Make use of the infrastructure to integrate people and *moliceiros*
- Expand the Aveiro Tech City Living Lab project

Architecture of the project



Data Acquisition

Detection Module (I)

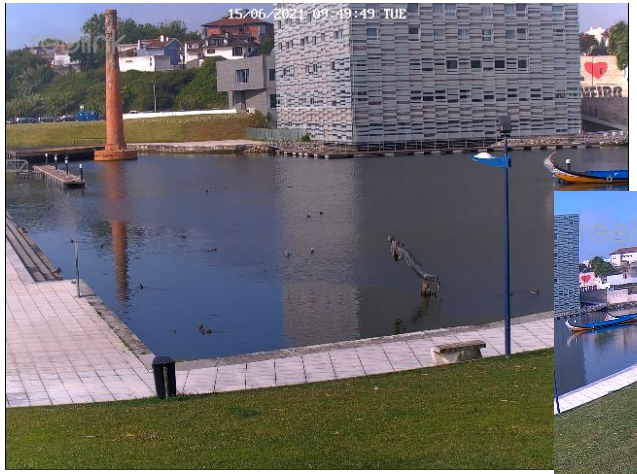
- Use of a Nvidia Software Development Kit, *DeepStream*, installed in two Jetson Nanos
- Video streaming analysis to detect people, vehicles, two wheeler vehicles and moliceiros
- Data sent to local brokers and then received by IT's central MQTT broker to be persisted
- The web application backend receives the data and displays it in the dashboard

Detection Module (II)

Problem	Solution
Loss of frames due to clock discrepancies	Change of the video sink used to process video frames
Values very high due to noise in the image	Search of false detection patterns to filter unwanted areas and adjustment of the threshold
No *DeepStream* model to perform *moliceiros* detection	Use of python library OpenCV and different detection module
Impossibility of using OpenCV and DeepStream at the same time in the same Jetson Nano	To be solved in future work

Detection Module (III)

- Three different positions of the camera



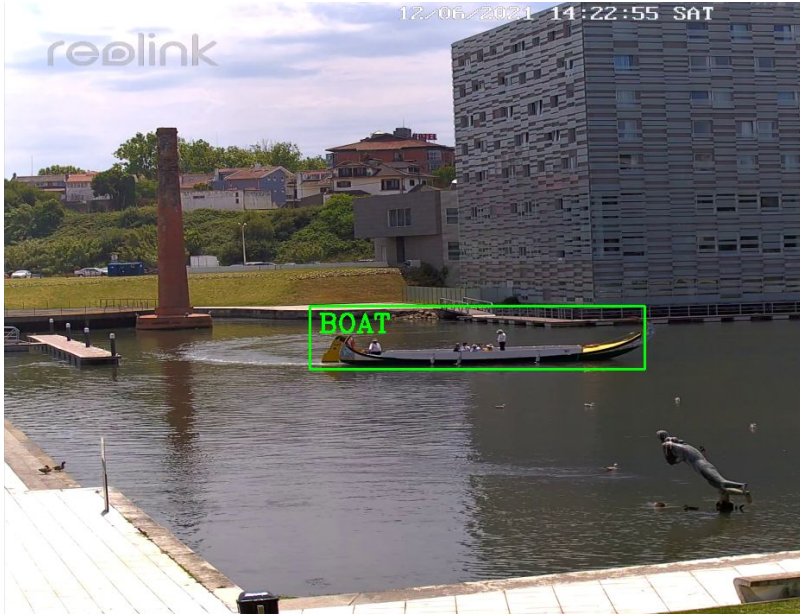
Moliceiros + People position

People position

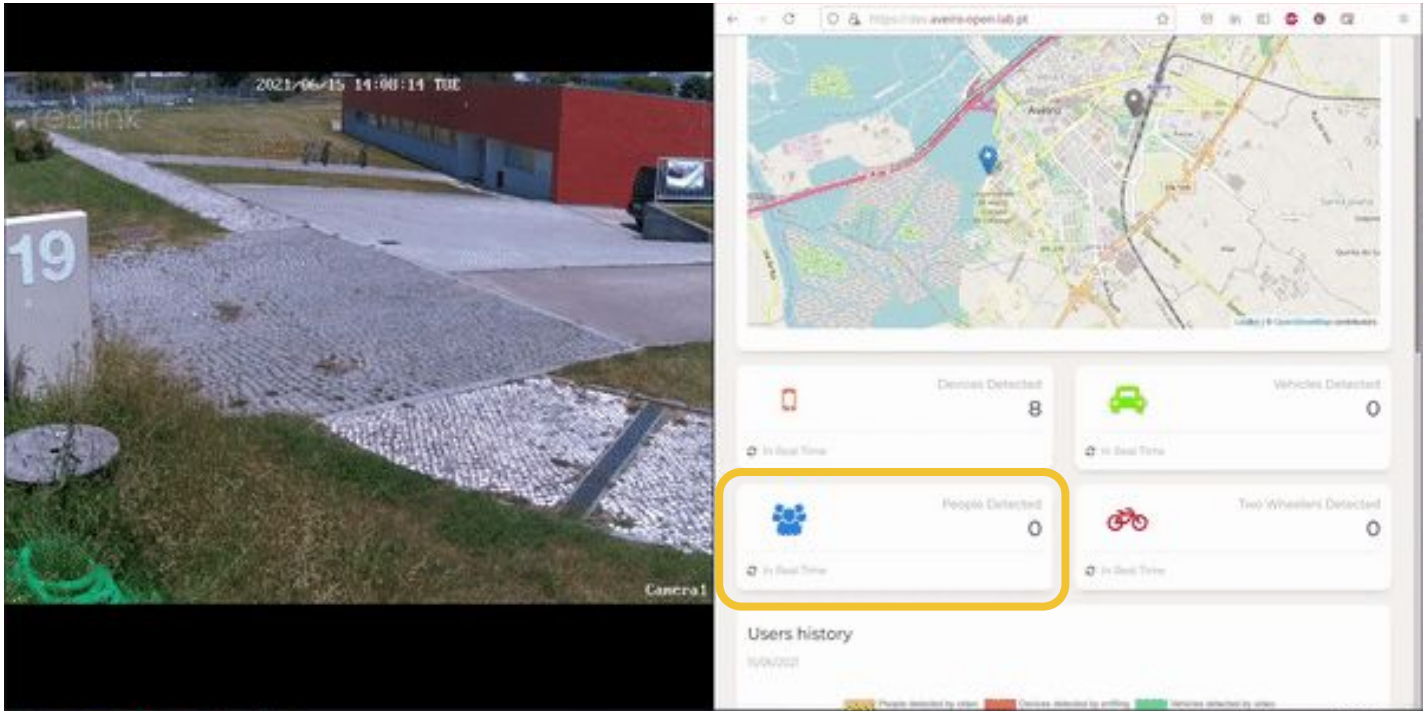
Cars + People position

Detection Module (IV)

- Detection of people, vehicles, two wheeler vehicles and moliceiros
- Frame processing of 30 FPS



Detection Module (V)



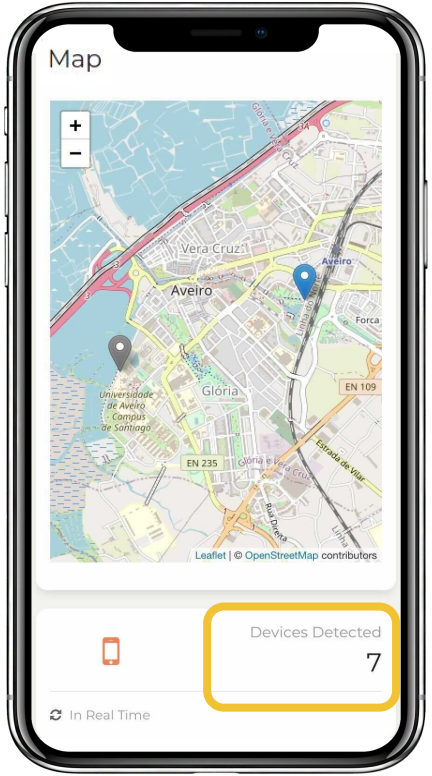
WiFi Detection Module (I)

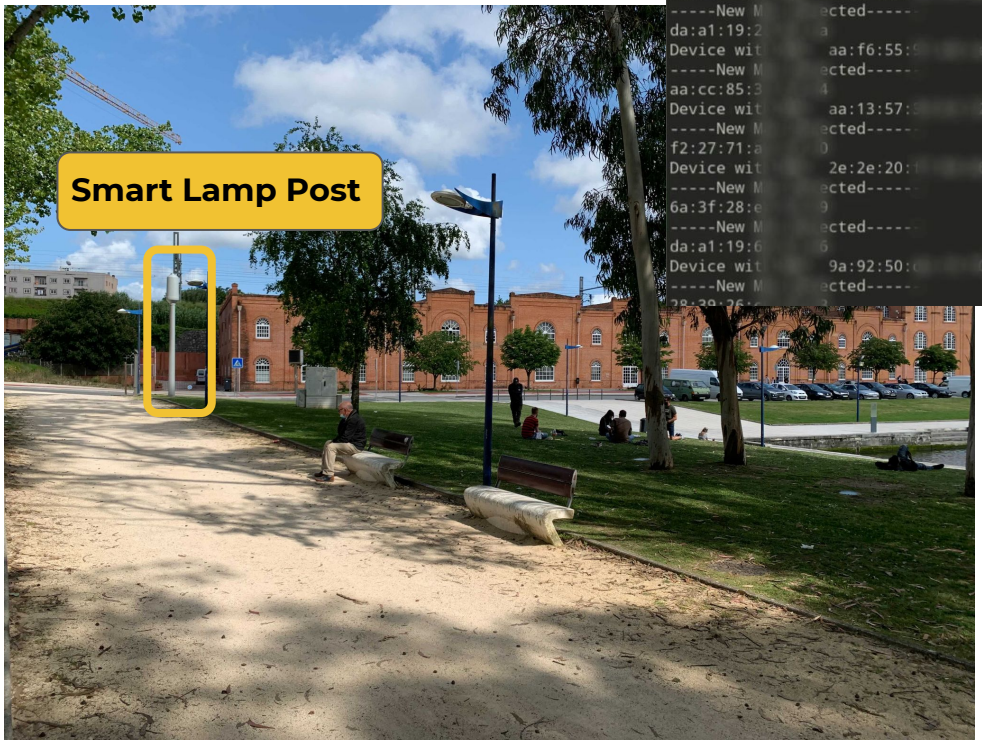
- Detection of devices near the smart lamp posts through capture of WiFi packets
- Distinction of different devices through the MAC address
- Estimation of the current number of devices in real time as well as the number of different people that pass near the Smart Lamp Posts
- Send data to the central broker in order to be persisted and shown in the web application

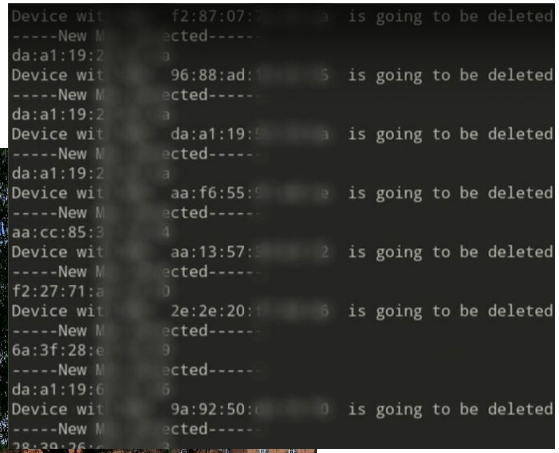
WiFi Detection Module (II)

Problem	Solution
The sniffing module used 100% of the APU's CPU which led to a crash	Use of another program to capture and filter packets in order to decrease the CPU usage: *pyshark*
Routers probe requests were being counted as devices	Excluded packets that contained an SSID and that were not broadcasted
Very high number of devices detected	Adjusted the detected packages time to live to obtain more accurate results

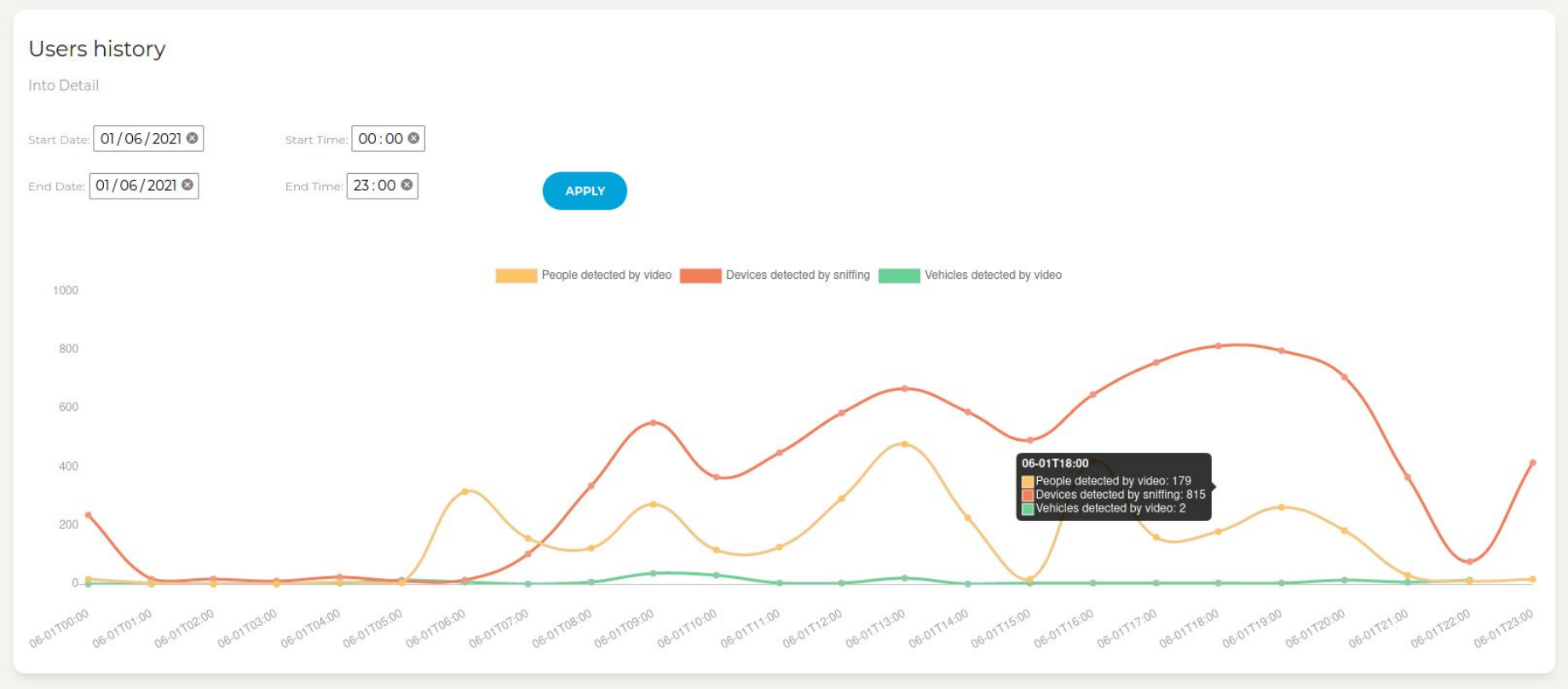
WiFi Detection Module (III)







Captured data (Before calibration)



Captured data (After calibration)

Sensor information BOT Hoje às 18:43

[Sniffing_p1] Avg: 20.9

[Sniffing_p22] Avg: 31.1

[Sniffing] More than 40 unique devices! Current Devices: 8

[Sniffing_p1] Avg: 37.5

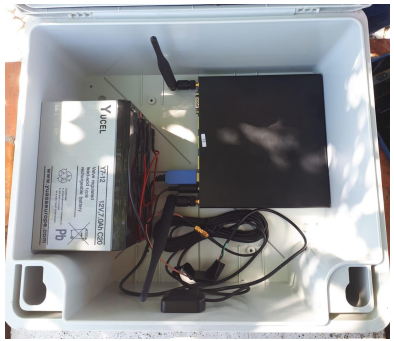
[Sniffing_p22] Avg: 30.8

[P22 Object Detection] More than 25 people!2

[Sniffing] More than 40 unique devices! Current Devices: 5

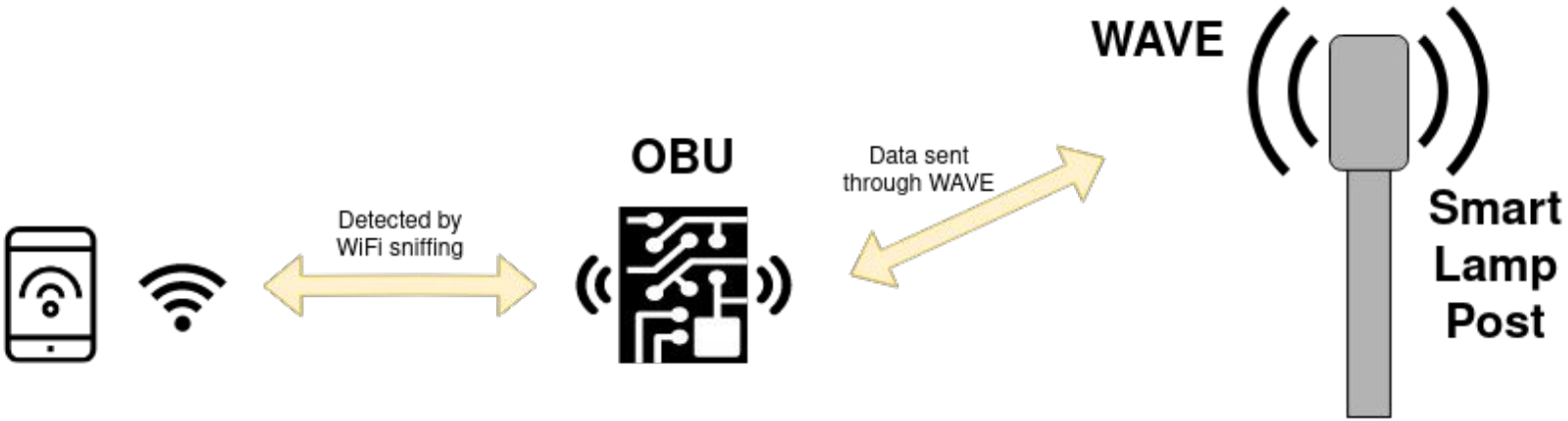
OBUs (On-Board Units) (I)

- Estimation of the number of people inside *moliceiros* using counting of devices through WiFi Detection
- 1 OBU includes 2 batteries, 1 APU and 1 GPS module
- Real time visualization of the location of the *moliceiros* via the GPS coordinates



OBUs (On-Board Units) (II)

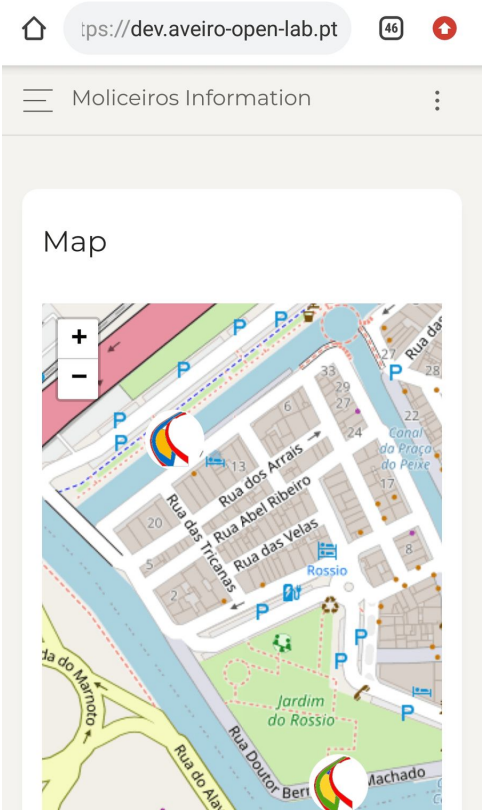
- OBUs in *moliceiros* do not have direct connection to the ATCLL's network
- Using the smart lamp posts near the *Ria de Aveiro* (RSUs) the OBUs send data through WAVE to the central server



OBUs (On-Board Units) (III)

Problem	Solution
Low Signal Strength	Repositioned the OBU to a higher open place in the moliceiros
Higher than expected battery consumption	Placed two batteries in parallel per OBU
Very high number of devices detected	Adjusted the detected packages time to live to obtain more accurate results

OBUs (On-Board Units) (IV)



Web Application

Web Application - Goals

- Display the collected data from the sensors in real-time
- Display *moliceiros* moving in a map based on the GPS location
- Show past-time data from the sensors
    - Based on user's input
    - Last 24 hours

Web Application - Back End

Back End - Problems

Problem	Solution
Web Sockets didn’t send any data	Use of Server Sent Events (SSEs)
SSEs have a maximum of 6 events at a time	Optimization of the number of SSEs used
Access to the cameras API was limited because it was closed-sourced	Reverse engineering of the camera’s API to access its controls from the Back End

Back End - Results

- Back End service : Flask
- Subscription to the central broker topics to obtain the real time data
- Access to an external API to obtain persisted data
- Development of an API to simplify and facilitate the access to information
- Use of Server Sent Events (SSE) to send the data to the dashboard
- Implementation of an account system, to manage access to certain information

Web Application - Front End

Live demonstration

Achieved goals

- All initial goals were achieved!

Extra goals achieved

- Use of OBUs (On-Board Units) in the middle step between the smartphones and the edge devices
- OBUs with Wi-Fi sniffing
- Estimation of the number of people inside a *moliceiro* by detecting how many devices are in proximity

Future Work

- Increase the accuracy of the number of devices detected through Wi-Fi Sniffing
- Implement the detection of *moliceiros*, people and vehicles using the same model and coordinating with the different positions of the camera
- Implement the data acquisition in other smart lamp posts in the city