# Smart Bike

Benat Ramirez, Merixtell, Vigo, Mark Suvorov, Group 7

**bicycle:**

- sensors:
    - GPS, speed
    - light sensor
    - temperature sensor

- actuators:
    - light
    - sound
    - remote lock, proximity lock
    - heater

- project:
    - mobile app
    - security
    - wifi

- functionalities:
    - proximity lock
    - parking search mode
    - autolights
    - position, speed and statistics
    - app, wireless

# 1  Introduction

The project MUST implement the following functionality:

1. **Sensor Data Collection:**
    - Collect data from sensors with the Raspberry Pi Pico W.

2. **Data Processing:**

   - Implement data processing on the Pico W.

3. **Output Control:**

   - Control output devices based on processed sensor data.
   - Result can be displayed with, e.g., LED, OLED, or Android/IOS App.

Further, the project MUST implement AT LEAST TWO of the functionalities below:

1. **Wireless Connectivity:**

   - Wireless communication with MQTT or HTTP should be implemented. Wi-Fi and Bluetooth can be for wireless connectivity. Pico W has onboard Wi-Fi module available. Bluetooth requires an external module.

2. **Cloud integration:**

   - Integrate with cloud platforms such as Infuxdb, Mangodb, AWS IoT or Google

3. **Mobile App:**

   - Develop Android/iOS applications for real-time data visualization and control, utilizing MQTT broker for sending and receiving data. Simple tools like MIT app inventor can be utilized for that purpose

4. **Security:**

   - Incorporate data encryption

5. **Edge Machine Learning:**

   - Implement edge-based machine learning models on the Pico W.

In this section, describe your application, and which of the above functionalities it implements. Provide the overall view of the application area, zooming in the particular application you've implemented.

# 2 Architecture

Describe here, with illustrations, your applications's sensor layer, networking layer, and data management layer.

# 3 Methods & Tools

Provide here a detailed description of the implementation of the system functionality, along with any libraries, tools, and methods used.

# 4 Evaluation

Provide here a detailed description of how you evaluated the application. Evaluate your implementation using selected criteria, for example:

- System throughput (e.g. number of messages).
- System latency (e.g., network and request processing latency.

Design two evaluation scenarios that you compare with each other, for example:

- Small number / large number of messages
- Small payload / big payload

# 5 Results

Collect numerical data of test cases:

- Collecting logs of container operations
- Conduct simple analysis for documentation purposes (e.g. plots or graphs)

# 6 Discussion

Discuss the results of the evaluation. What weaknesses does your application have? In what scenarios does it work well? Return back to the application area you described in Introduction – put the results in context.