SMART SWIMMING POOL

MATTEO MUGNAI,
FILIPPO PUCCINI

INTRODUCTION

Use case: Smart IoT and telemetry system for the control and management of a **swimming pool**

Main goal: simulate a LLN composed by sensors and actuators capable of monitor and automatically react to changes in environmental conditions through a control system handled by a JAVA application

IOT DEVICES

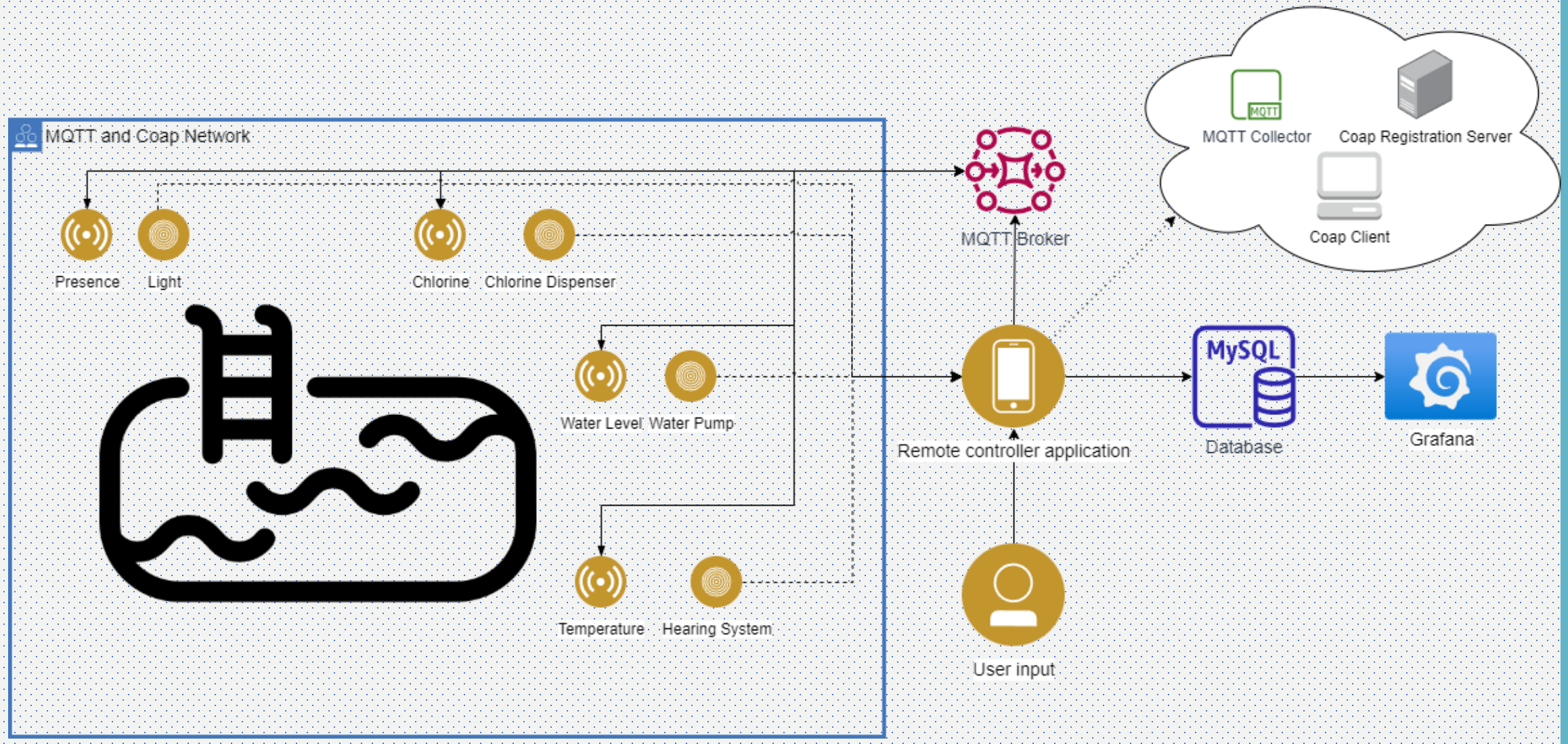
Sensors (MQTT)

- Presence (bool)
- Water temperature (int, °C)
- Water level (int, %)
- Chlorine level (int, %)

Actuators (CoAP)

- Lights (ON|OFF)
- Heating system (INC|DEC|OFF)
- Water pump (INC|DEC|OFF)
- Chlorine dispenser (ON|OFF)

SYSTEM ARCHITECTURE



COAP

Light

CLIENT: Cloud App

SERVER: IoT actuator

- /light/switch?mode=['ON', 'OFF']
- /light/color?color=['r', 'g', 'b']

→PUT: method to change the status or the color of lights

Heating system

CLIENT: Cloud App

SERVER: IoT actuator

- /heating_system/switch?mode=['INC', 'DEC', 'OFF']

→PUT: method to change the status of heating system

Chlorine dispenser

CLIENT: Cloud App

SERVER: IoT actuator

- /chlorine_dispenser/switch?mode=['ON', 'OFF']

→PUT: method to change the status of chlorine dispenser

Water pump

CLIENT: Cloud App

SERVER: IoT actuator

- /water_pump/switch?mode=['INC', 'DEC', 'OFF']

→PUT: method to change the status of water pump

Actuator registration

CLIENT: IoT actuator

SERVER: Cloud App

- /registration

→ POST: method to add a new actuators in the database

→ DELETE: method to delete an actuator from database

MQTT

Measurements

PUB → IoT device

SUB → Cloud App

- Presence (topic = ‘presence’)
- Chlorine (topic = ‘chlorine’)
- Temperature (topic = ‘temperature’)
- Water_level (topic = ‘water-level’)

These topics are used to periodically (every 1s) send to cloud app the values sensed by devices

Sensor notification

PUB → Cloud App

SUB → IoT device

- Presence (topic = ‘light-command’)
- Chlorine (topic = ‘chlorine-command’)
- Temperature (topic = ‘temperature-command’)
- Water_level (topic = ‘water-level-command’)

These topics are used to inform a specific sensor that an actuator has been activated (to trigger a change in sintetic data generation from ficticious sensors)

DATA ENCODING (JSON)

WaterLevel →

```
{"nodeId": "001",
"height": "50"}
```

Presence →

```
{"nodeId":“002",
"presence":“true"}
```

Temperature →

```
{"nodeId":“003",
"temperature":“25"}
```

Chlorine →

```
{"nodeId":“004",
"chloirne":“75"}
```

All sensors return the data they have collected in JSON format:

- JSON is more flexible and less verbose than XML
- JSON is lighter and faster to process than XML
- Our application does not process critical data

CLOUD APPLICATION

- Implemented in JAVA using *Californium* and *Paho* library
- Stores data coming from sensors in a MySQL DB
- Stores all CoAP messages in a log file
- Allow user interaction through a simple command interface:
  - Get last measurement sensed
  - Manually intervention on actuators (ON/OFF or INC/DEC)
  - Set new thresholds (lowerBound and upperBound) in which a specific measure must stay
- Handles a control logic to regulate actuators when it's necessary:
  - Trigger activation and deactivation of actuators when values are not in the correct range

CONTROL LOGIC

Temperature:

- When the average of the last 5 samples received is above/below a threshold we send a CoAP message to activate heating system in 'DEC'/'INC' mode

WaterLevel and ChlorineLevel:

- When we detect three consecutive samples above/below a certain threshold we send a CoAP message to activate water pump ('INC' / 'DEC') or chlorine dispenser ('ON')

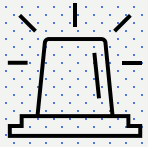
Presence:

- When we receive a 'true' value we send a CoAP message to switch 'ON' lights

Actuators remain active until the level of the associated measure come back to the exact mean between lowerBound and upperBound

BUTTONS & LEDS

Buttons have been used on actuators:



-HeatingSystem

-WaterPump

-ChlorineDispenser

→Press button to switch ON/OFF the relative actuator

Leds have been used on actuators:

-HeatingSystem

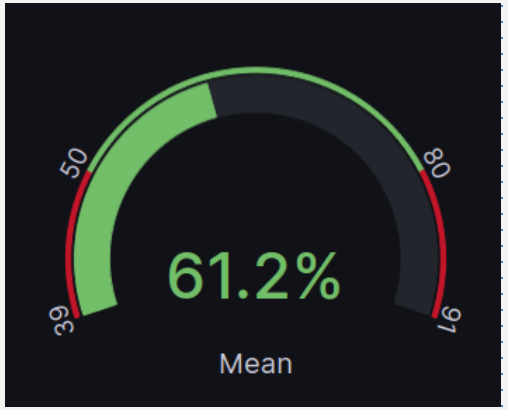
-WaterPump

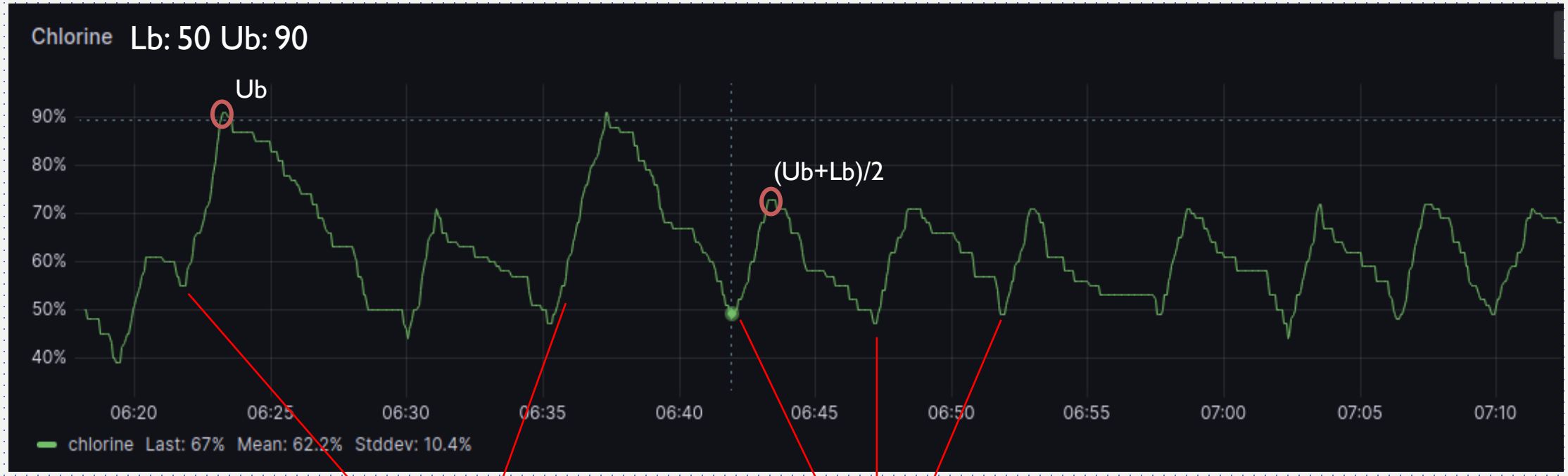
-ChlorineDispenser

'GREEN' led means actuator ON, 'RED' led means actuator OFF

-Light → led ON indicates the current colour of the light ('GREEN','RED','YELLOW')

GRAFANA



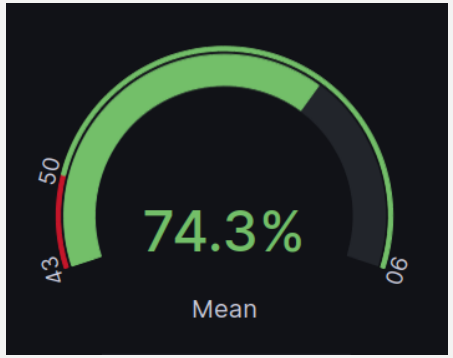


Pressed button manually

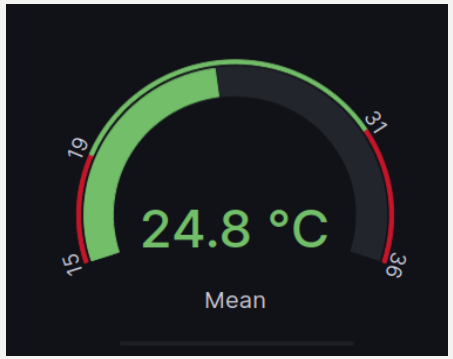
Trigger regulates level automatically

GRAFANA


Water Level
90%
80%
70%
60%
50%
40%
06:20
06:25
06:30
06:35
06:40
06:45
06:50
06:55
07:00
07:05
07:10
height Last: 71% Mean: 74.3% Stddev: 8.66%


50
43
90
74.3%
Mean


Temperature
35 °C
30 °C
25 °C
20 °C
15 °C
05:15
05:20
05:25
05:30
05:35
05:40
05:45
05:50
05:55
Mean Last: 30 °C Mean: 24.8 °C Stddev: 3.40 °C


19
31
15
36
24.8 °C
Mean