SmartSauna

FEDERICA BALDI – DANIELE CIOFFO

Introduction

The benefits of the sauna for the human body are well known but, to take full advantage of them, it is essential that a proper environment is maintained.

Our goal: to create a smart solution for the management of a sauna, automating all the procedures through IoT technology.

Temperature

Humidity

CO2 Concentration

Number of People

Smart lighting and chromotherapy

System Architecture

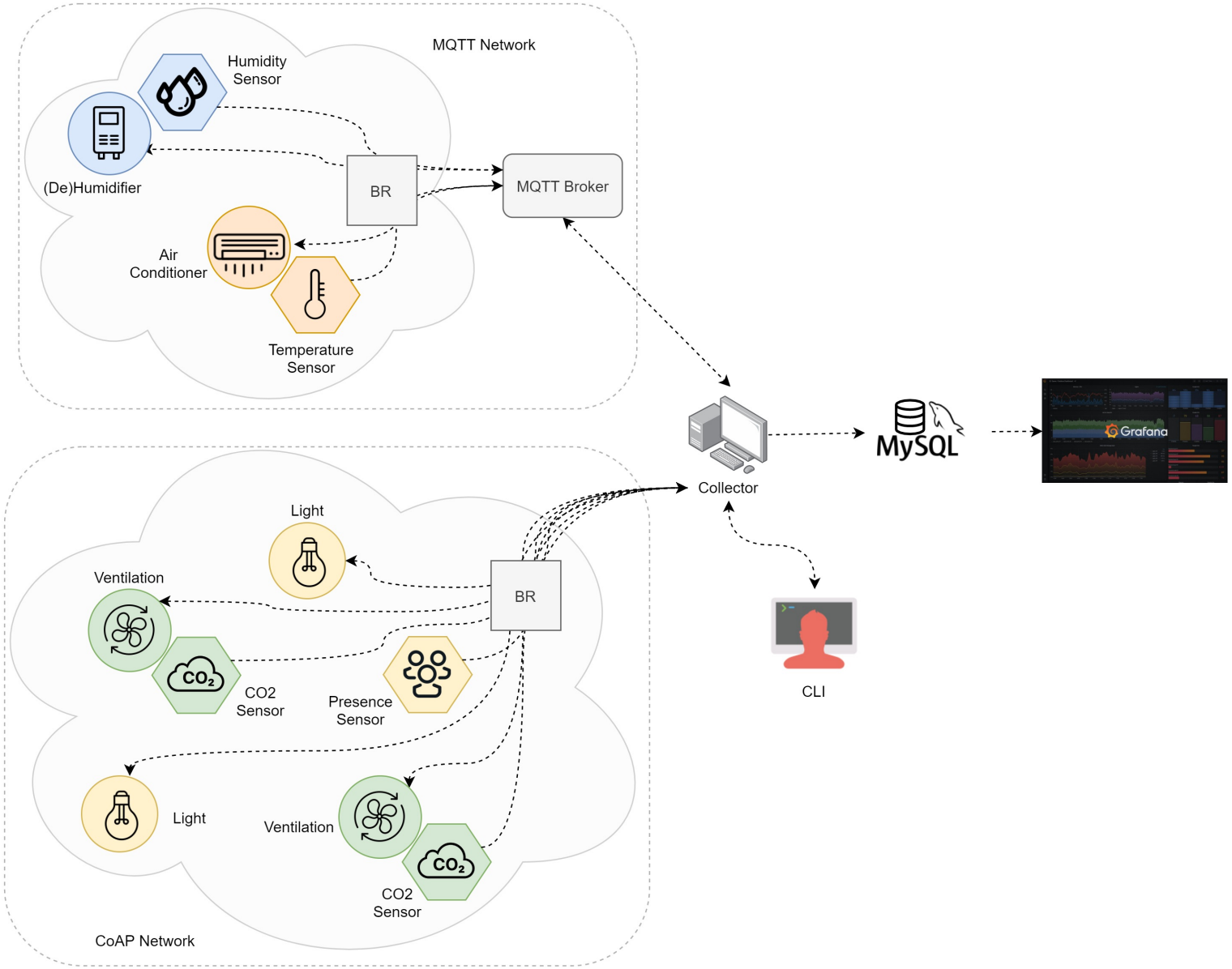
Two networks of IoT devices:

- MQTT network
  - Humidity Sensor + (De)Humidifier
  - Temperature Sensor + Air Conditioner
- CoAP network
  - Presence Sensor
  - Two *smart* Lights
  - Two CO2 Sensors + Ventilation System

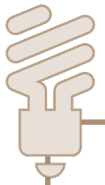
A Java **Collector** that:

- Receives data from sensors
- Executes control logic
- Writes to a MySQL database
- Exposes a CLI

A **web interface** (developed using Grafana)

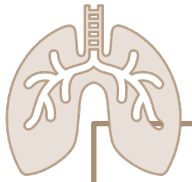


CoAP



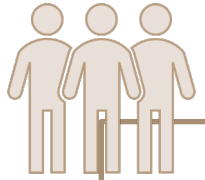
Light

- /light/switch
  - → PUT mode=ON/OFF: turn on or turn off the light
- /light/color
  - → PUT color=GREEN/YELLOW/RED: change the color of the light



Air Quality

- /air_quality/co2
  - → GET: get the current CO2 concentration (ppm)
  - → EVENT: notify observers of a change
- /air_quality/ventilation
  - → PUT mode=ON/OFF: turn the ventilation system on or off

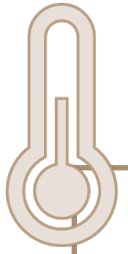


Presence

- /presence
  - → GET: get the number of people currently in the sauna
  - → PUT “maxNumberOfPeople”: set a new maximum number of people
  - → EVENT: notify observers of a change

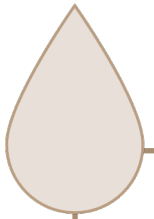
MQTT

Temperature



- **temperature** (pub = device, sub = controller): publish temperature (°C) measurements periodically
- **AC** (pub = controller, sub = device): enable/disable the actuator
  - → **INC:** increase the temperature
  - → **DEC:** decrease the temperature
  - → **OFF:** turn off the air conditioner

Humidity



- **humidity** (pub = device, sub = controller): publish humidity percentage measurements periodically
- **humidifier** (pub = controller, sub = device): enable/disable the actuator
  - → **INC:** increase the humidity level
  - → **DEC:** decrease the humidity level
  - → **OFF:** turn off the (de)humidifier

Data Encoding

```
{
   "node":"nodeID",
   "concentration":"co2Concentration"
}
```

```
{
   "node":"nodeID",
   "humidity":"humidityLevel"
}
```

```
{
   "node":"nodeID",
   "temperature":"temperature"
}
```

```
{
   "node":"nodeID",
   "quantity":"numberOfPeople"
}
```

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

- JSON is more flexible and less verbose than XML
- our application does not process critical data
- libraries for CBOR are very recent and not yet complete