Smart Gallery

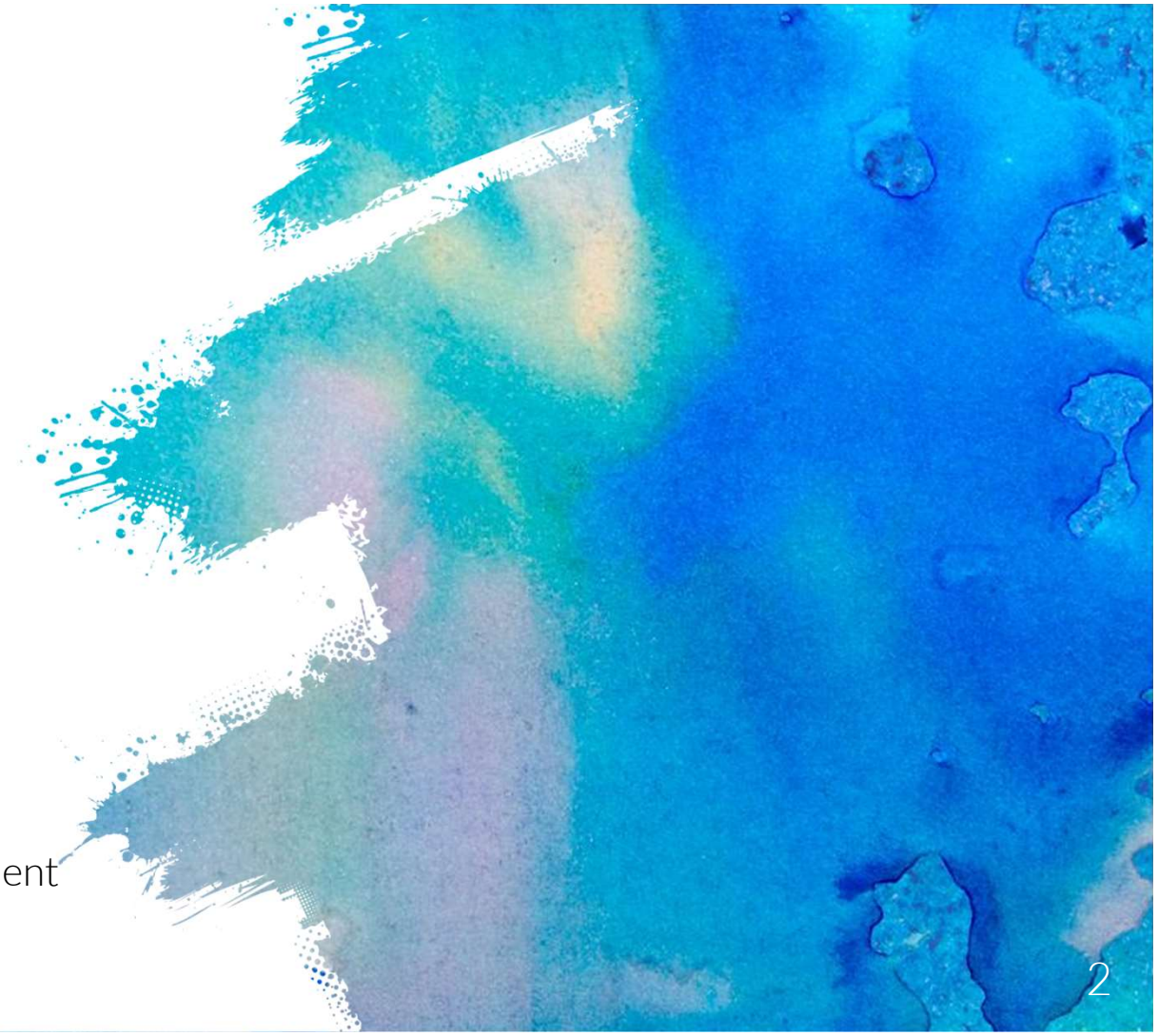
Programming for IoT applications

Supervisors:
PhD. Andrea Acquaviva
PhD. Edoardo Patti

Carla Corona
Navid Yamini
Ximena Garzón
Sadegh Bibak

Outlines

1. Motivation
2. Tools Used
3. General Schema
4. Resource Catalog
5. Services on the Raspberry Pi
6. MQTT to Web Service
7. Threshold monitoring
8. ThingSpeak and the Adapter
9. Telegram
10. Artwork Info. Web Service
11. Artwork Info. Application
12. Movement Detection
13. Web Page GUI
14. Issues and Future Improvement



1. Motivation

Biggest challenges of museums/galleries:

- *"**Buildings** - although extensive reconstructions historical buildings are often compromise of security, daily operation, not enough space and design of exhibitions…"[1]*
  "Our research into historical and cultural alternatives, our commitment to public outreach [illegible] compliance, and our infrastructure adaptations [illegible] as leaders in the drive toward sustainability"[2]
- *"**Engaging Interpretation**:.. Evolving exhibition techniques allow adapting museums to present multiple viewpoints and to educate and challenge visitors as never before"[3]*

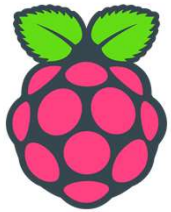
Smart Gallery

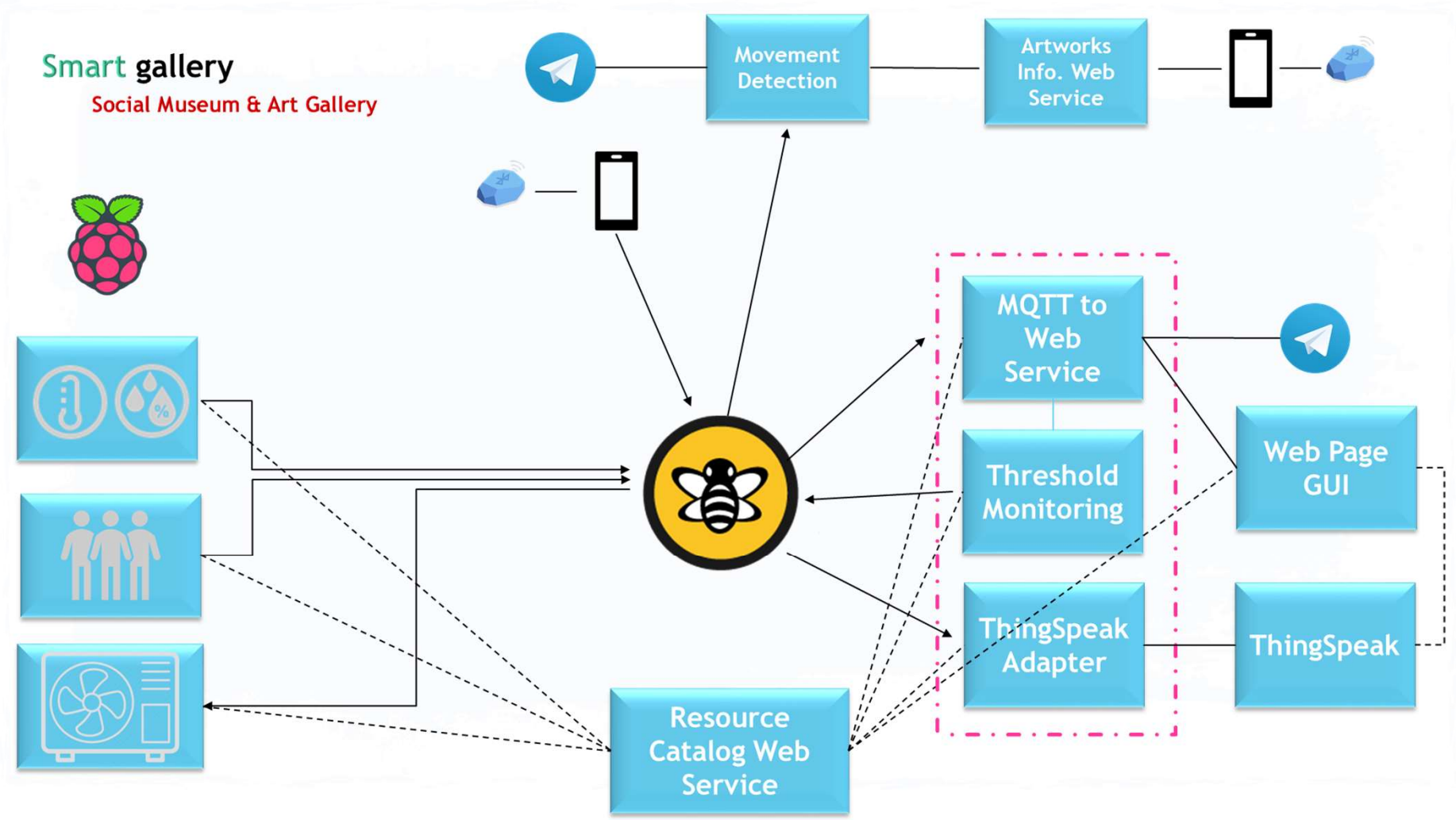
Social Museum & Art Gallery

[1]" What are the biggest challenges facing museums today?",, Bryan Rayca, URL: https://www.quora.com/What-are-the-biggest-challenges-facing-museums-today
[2]" The Next Sustainability Frontier", American Alliance of Museums, Museum2040 magazine, pag. 12. URL: https://www.aam-us.org/wp-content/uploads/2018/04/museum2040.pdf
[3]" Challenges Museums Currently Face" , Verner Johnson, URL: http://www.vernerjohnson.com/approach/

How did we do it?

2. Tools that we used




Smart gallery
Social Museum & Art Gallery
Movement Detection
Artworks Info. Web Service
MQTT to Web Service
Threshold Monitoring
ThingSpeak Adapter
Web Page GUI
ThingSpeak
Resource Catalog Web Service

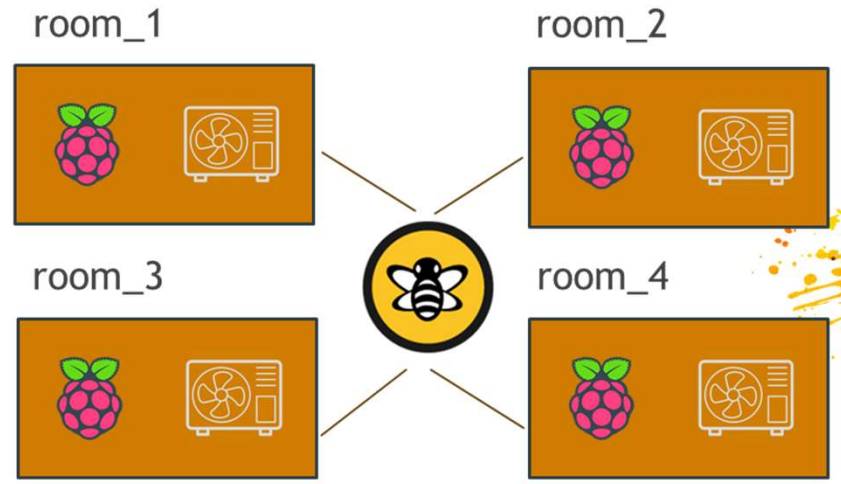
4. Resource Catalog Web Service

- Expose parameters for the different services

```
{
    "broker": {
        "Broker_IP": "192.168.1.110",
        "Broker_port": "1883"
    },
    "telegram": {
        "Port": "371024597:AAGK5je2cAXhyf4oMMD5wcUj1SguoZC5ZOY",
        "chatID": "94432957"
    },
    "dataToRest": {
        "Host_IP": "192.168.1.125",
        "port": "8082"
    },
    "room_1": {
        "topic": {
            "Ac_Status": "Gallery1/Room1/Status",
            "AC_Topic": "Gallery1/Room1/Order",
            "DHT_Topic": "Gallery1/Room1/TempHum1",
            "Counter_Topic": "Gallery1/Room1/BlueTooth1"
        },
        "thresholds": {
            "min_hum": "0.0",
            "min_temp": "0.0",
            "max_hum": "10.0",
            "max_temp": "10.0"
        },
        "thingspeak": {
            "READ_API_KEY": "PN6SHE0RIBDLGS85",
            "ACCESS_TOKEN": "DDNTX8BUX8A17YZG",
            "tTransport": "websockets",
            "channelID": "240810",
            "tPort": "80",
            "mqttHost": "mqtt.thingspeak.com",
```

length : 1,923 lines : 65 Ln : 59 Col : 31 Sel : 0 | 0 Windows (CR LF) UTF-8

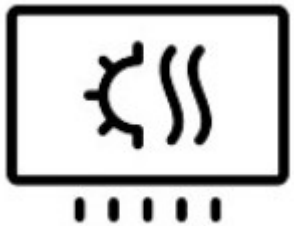
4. Network Assumption




```
{
    "reSourceCatalog": {
        "url": "http://192.168.1.113:8080/",
        "roomId":"room_1"
    }
}
```

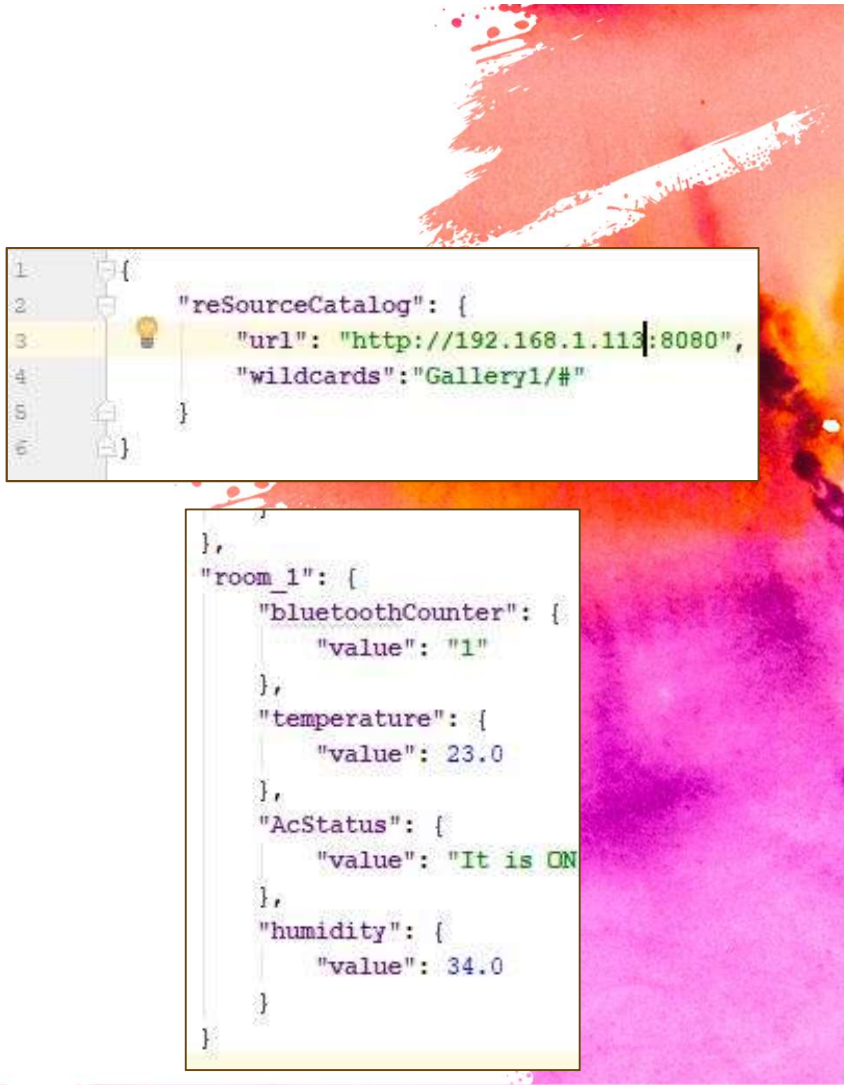
5. Services on Raspberry Pi

- Temperature and Humidity
  - Publishes the data sensed by temperature and humidity sensor
- Counting Number of People
  - Counting the enabled Bluetooth devices
  - Publish the counted number to the broker
- Subscribe Air conditioner order
  - Subscribe the orders coming from threshold checking
  - Apply the order in case it is different from the current situation
  - Publish the new status of the air conditioner



6. MQTT to Web Service

- Subscribe data from the broker through topics
  - Temperature and Humidity
  - Number of Bluetooth device
  - Order for the A/C
  - Status of the A/C
- Update data in the real time data JSON file
- Get the URL from resource catalog
- Expose the updated file to other services

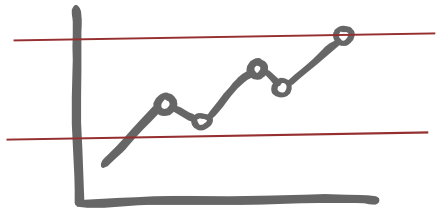


7. Threshold monitoring

Get the real time data from MQTT to web Service

Get the thresholds from resource catalog for the specified room

Publish the A/C order



8. ThingSpeak and its adapter

- Subscribe data from the broker through topics
  - Temperature and Humidity
  - Number of Bluetooth device
  - Order for the A/C
  - Status of the A/C (0: OFF, 1: ON)

- Publish it on the ThingSpeak channels

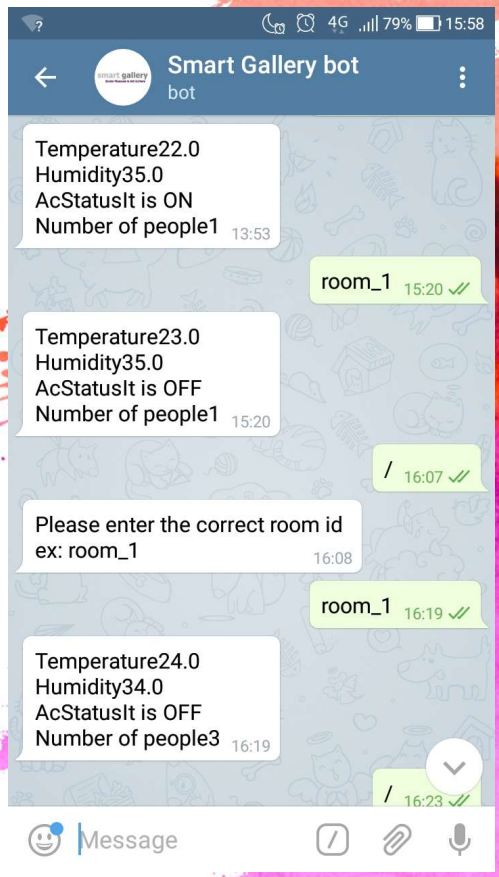
9. Telegram

Wait on the listening mode

Get the room ID

Send back the data

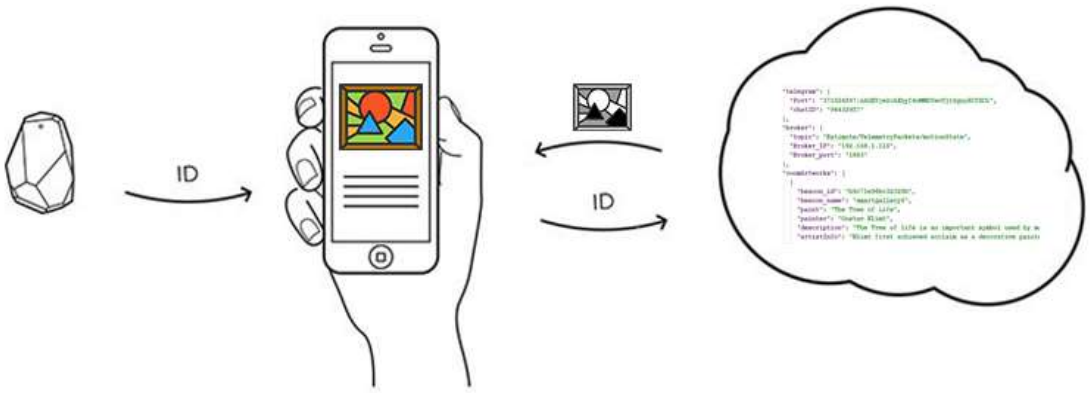
- Temperature
- Humidity
- Number of Victors at the moment
- A/C Status



10. Artwork info. web service

Estimote Beacon sensor

- Communicate through Bluetooth
- Several types of packets (iBeacon)
- Send information (ID and contents)
- Range ~ 40 – 50m



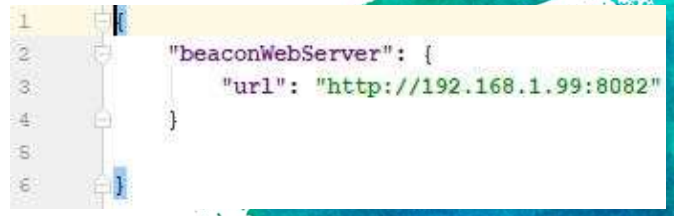
- Allows manager to set the data related to this service through win.app (DataEntry)
  - -Telegram information (Port, chatID)
  - -Broker parameters (Topic, IP, Port)
  - -All artwork information that must be set inside the application
- A JSON file is created with these parameters
- The file is exposed through a REST service

11. Artwork info. application

- Read the configuration file to find the address where the information is exposed

- Take the data from the Exposed JSON file
  - Telegram port and id
  - Broker port, IP and Topic
  - Artwork information

- Show the content
  - Based on the ID received
  - Broker port, IP and Topic
  - Artwork information

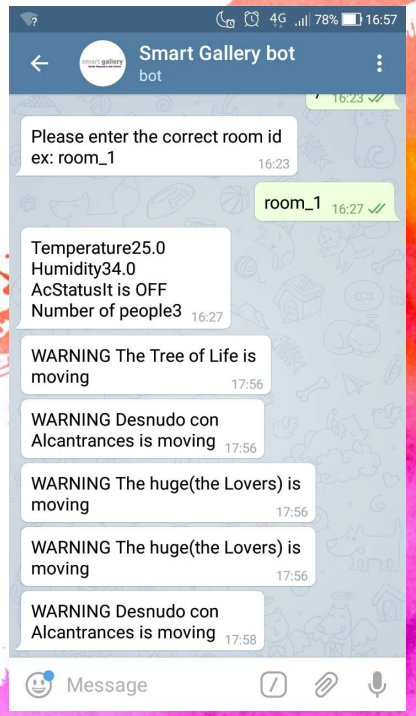


12. Movement Detection

- Beacons send 'Telemetry data' with information about the accelerometer
  - Motion status can be 0 or 1

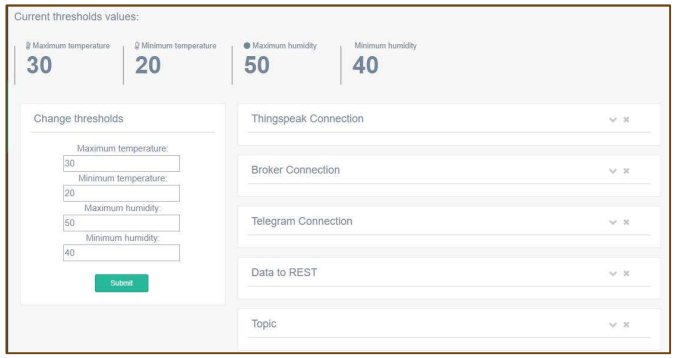
- An application on the monitoring device (iPhone) receive them
  - Receives these data
  - Publish them to the MQTT Broker

- A subscriber (implemented in a separate device)
  - Take the data
  - Send a warning message with the name of the artwork to Telegram



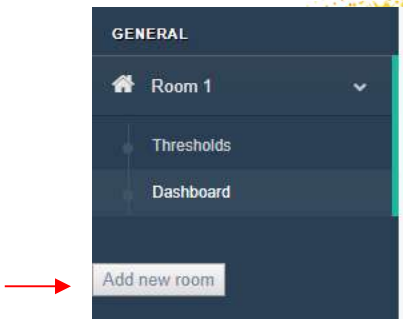
13. Web Page GUI

- Have access to data from both ThingSpeak and MQTT



- Can modify the thresholds, topics and general settings.

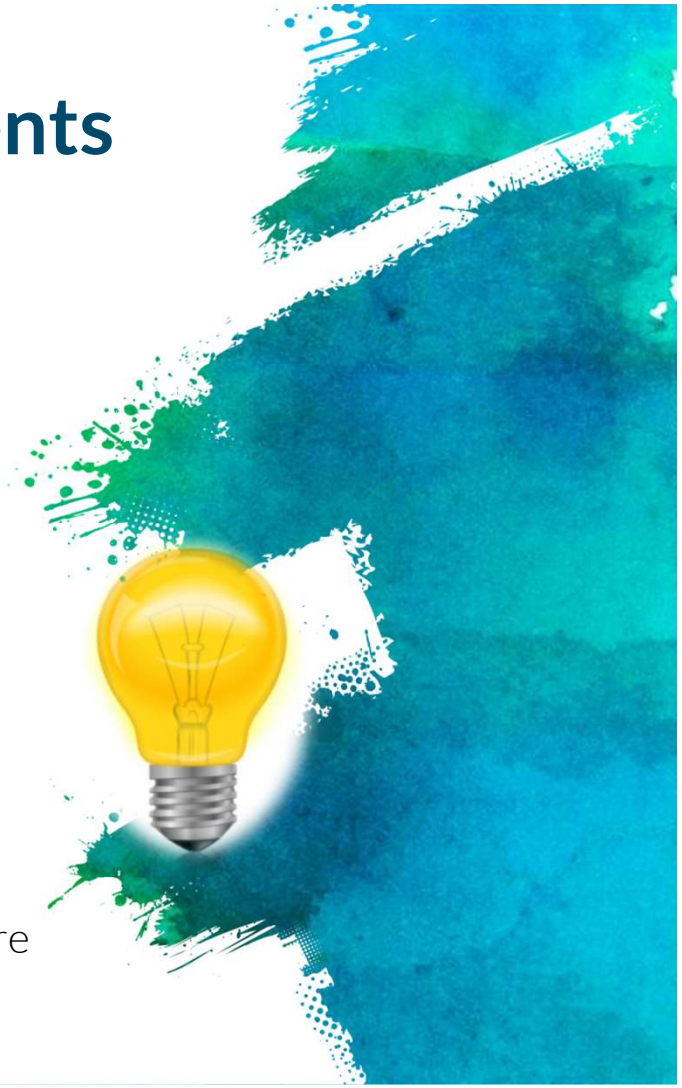
- Ability to add a new room



14. Issues and Future Improvements

- ThingSpeak has a rate limit
- Retrieving data from ThingSpeak takes time
- Previous documentation for what concerning beacons was difficult to manage and implement
- Dynamic allocation of the content related to the artworks without hardcoding it inside the app
- Keep the resource catalog and the webpage in 2 different servers.

- **Future Improvements**
  - Test the system with more raspberries.
  - Improve the mobile apps
  - Use of a HTML template for the webpage (Jinja2).
  - Offering the opportunity to integrate the managing of more physical quantity that can affect a museum environment




Thank you!
Grazie!
Gracias!
سپاسگزارم