



NextRoom

- Nicolò Palmiero
- Luigi Sigillo

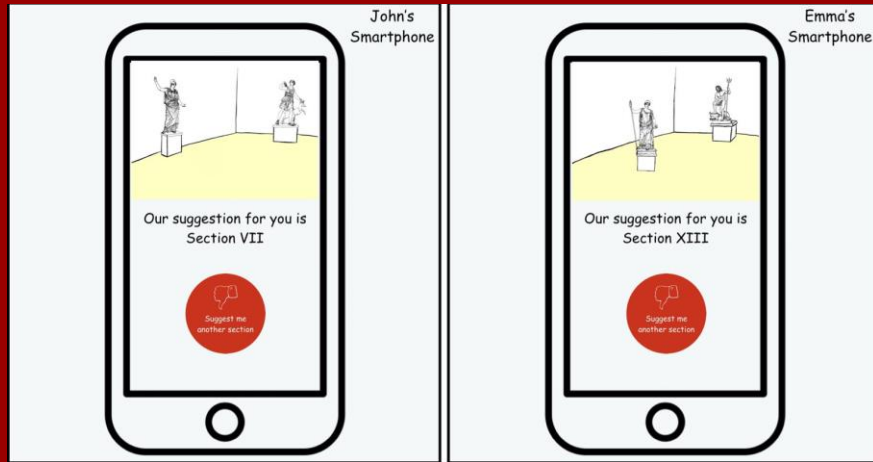
Supervisor: Joy Abi Rizk

Repository

<https://github.com/LuigiSigillo/lotBigProject>

The problems

- Personalized tours in the museum



- Museum visits during the COVID-19 emergency





Existing approaches

What is GEM?

A mobile app that creates a relationship between the museum and their audience, before, during and after their visit. GEM uses AI and big-data analytics to transform each tour into a **unique and memorable experience**

Visitor's Experience

Where are visitors lingering? receive deep statistics and demographics insights

Souvenir Shop

Customized suggestions for souvenirs, based on the visitor's tour

Marketing

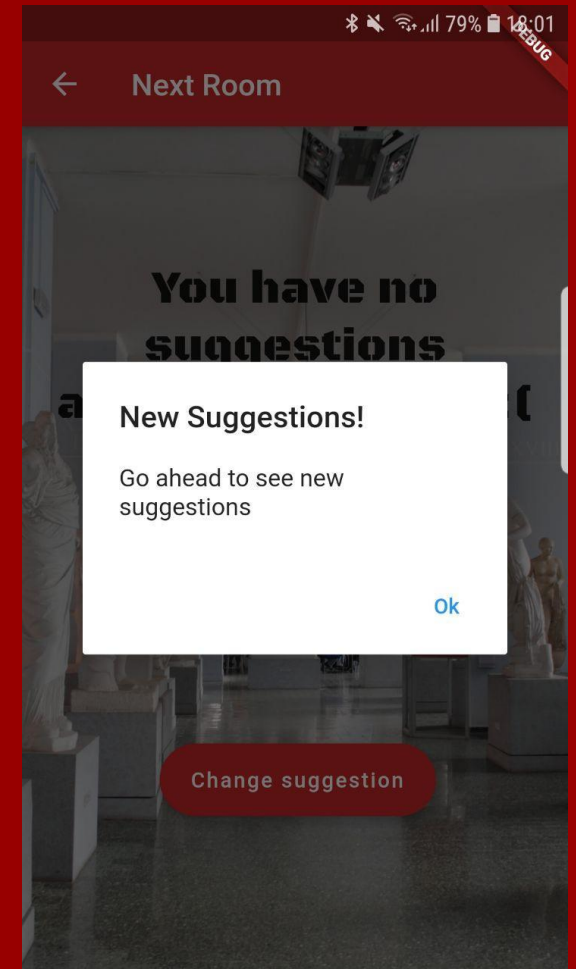
Expose visitors to additional activities (lectures, workshops, events)

Engagement

Encourage visitors to explore new exhibits, get a summary of their tour and share it

Our solution for personalized tours

- Mobile application available for both iOS and Android
- During a tour suggests the next section you could visit based on the time you have spent in the previous sections
- The application will display a preview of the suggested section
 - The user could choose to follow it or to jump to another suggestion.
- The suggestions are sent every time a user is leaving a section



Our solution to avoid gatherings

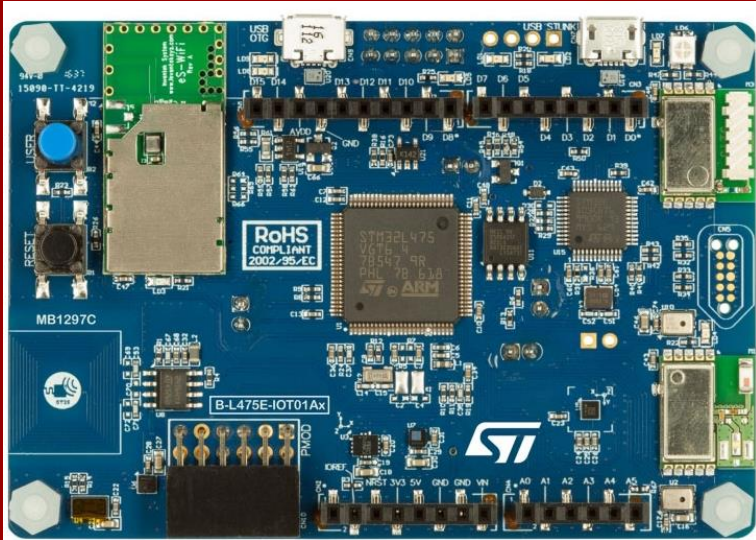
Solve the crowd problem of our personalized tour in these times in which sections cannot be overcrowded:

- Added features to try to equalize the number of people in each section.
 - Suggest the most interesting section for the visitor and, at the same time, try to avoid the formation of gatherings
- Estimating the proximity of two different visitor and check if they are respecting the security distance.



Hardware Components

B-L475E-IOT01A Discovery kit



- ▶ Ultra-low-power STM32L4 Series MCUs based on Arm® Cortex®-M4 core with 1 Mbyte of Flash memory and 128 Kbytes of SRAM
- ▶ Bluetooth® V4.1 module (BLE technology)
- ▶ 802.11 b/g/n compliant Wi-Fi® module from Inventek

arm MBED OS



Software
services

IoT components

- Eclipse Mosquitto as MQTT broker



- Azure IoT Hub



Azure IoT Hub

Cloud components

- Azure Function to calculate the suggestion



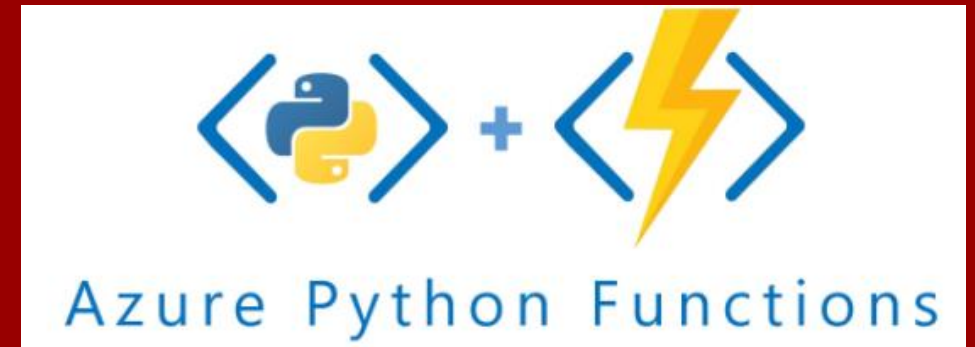
- Azure SQL DB to store the data



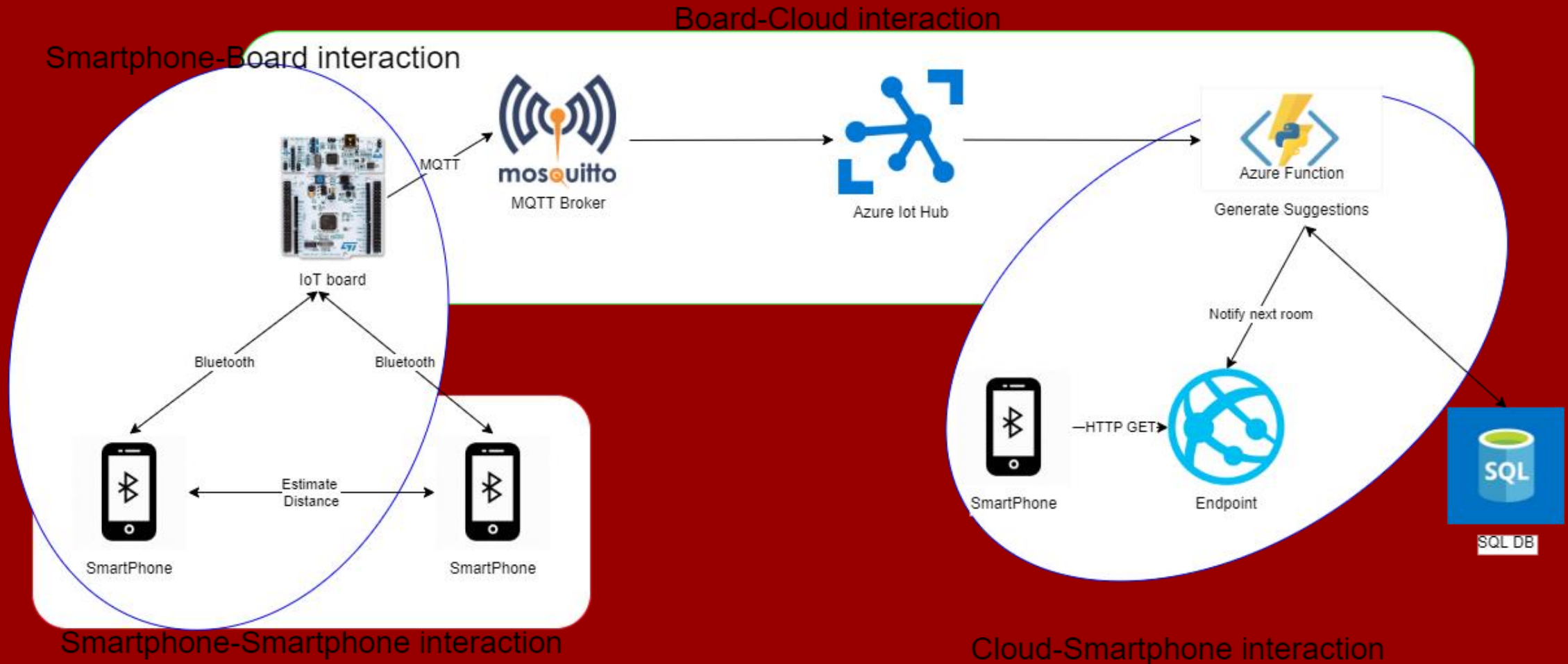
Programming languages used

- c++ to write the code for the IoT board using Mbed OS
- Python for the suggestion algorithm and the data persistence
- Dart to write the code for the mobile application using Flutter

arm MBED OS



Network diagram



The background is a solid red color. In the top-left corner, there is a blue parallelogram. To its right and slightly overlapping is a teal parallelogram. Below the blue shape, there is a lighter red parallelogram. The word "Evaluation" is written in white, sans-serif font on the right side of the image.

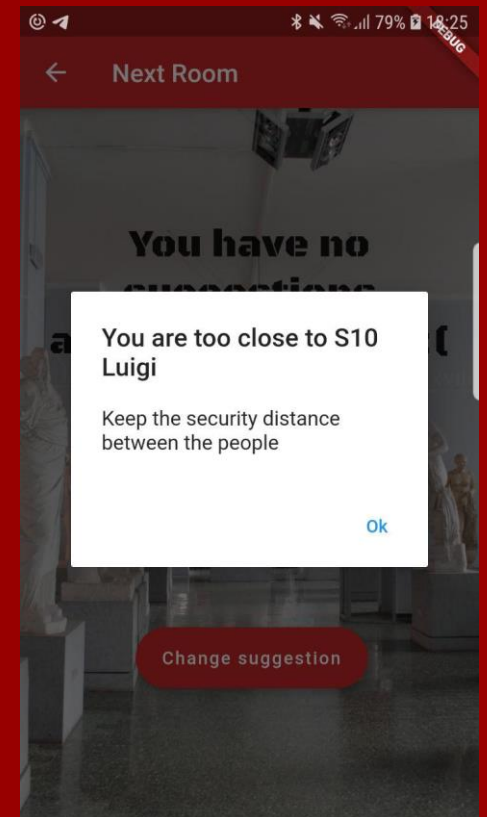
Evaluation

Technical Evaluation

- We have done a technical evaluation performing load tests on all parts of our system:
 - **BLE** IoT device-Smartphone interaction: how many smartphones a single device can handle?
 - **Cloud** IoT device-Cloud interaction: what is the message rate with which the device can send messages to Azure IoT hub?
 - We have performed a simulation, through a Python script, sending data to the cloud and analyzing the behaviour of our algorithm. We discovered that saving data to the DB, is an important bottleneck, so we have decided to collect the data from the board at least every 30 seconds after the previous data collection.

Technical Evaluation

- We have tested the feature that aim to calculate the distance between two visitor, as we expected is not too precise. This is since every vendor use different Bluetooth sensor on their smartphone, so the RSSI at one meter that is used to estimate the distance is not a unique value.
 - We decided to introduce a threshold, if we detect that two people are too near to each other for more than three times we would notice to them.



Pricing Evaluation

Board Pricing

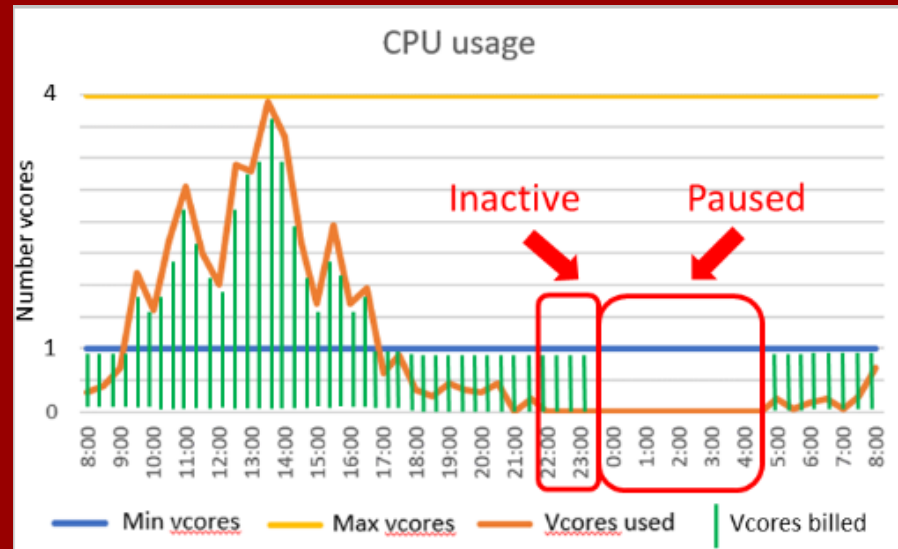
- The board used is the B-L475E-IOT01A Discovery kit that has a retail price of about 50€
- Considering at least two rooms per section, the estimated cost will be around 500€

Cloud Pricing

- Azure Function: the first 400,000 GB/s of execution and 1,000,000 executions are free. Then you pay what you consume (serverless)
- App service: The basic plan cost around 60€, for testing purpose we will use the free one.
- Azure SQL Database: We choose to use the serverless option also in the DB, we use the maximum size of 15GB but it is possible to use more space.

Pricing Evaluation

Microsoft Azure Estimate					
Your Estimate					
Service type	Custom name	Region	Description	Estimated monthly cost	Estimated upfront cost
Azure Functions		France Central	Consumption tier, 128 MB memory, 100 milliseconds execution time, 0 executions/mo	€0,00	€0,00
App Service		France Central	Basic Tier; 1 B1 (1 Core(s), 1.75 GB RAM, 10 GB Storage) x 730 Hours; Windows OS	€57,87	€0,00
Azure SQL Database		France Central	Single Database, vCore Purchase Model, General Purpose Tier, Serverless, Gen 5, 1 Billed vCores, 16 GB Storage, 0 GB Backup Storage	€2,49	€0,00
Support			Support	€0,00	€0,00
			Licensing Program	Microsoft Online Services Agreement	
			Total	€60,36	€0,00



UX evaluation

?

Future developments

1. Dashboard for the curators of the museum to monitor the number of people in the different sections of the museum.

Thank you for
listening