BlueFace Secure Lock

Giovanni Bartolomeo, Antonio Boffa, Elena Lepri, Jacopo Massa, Giulio Purgatorio

A.A. 2019/20

Context and problem description

In public buildings, such as workplaces or departments, it is important to control access to personal offices or other particular rooms.

For each area, it must be possible to set the authorized users and the access policy, possibly with different levels of security depending on the type of room.

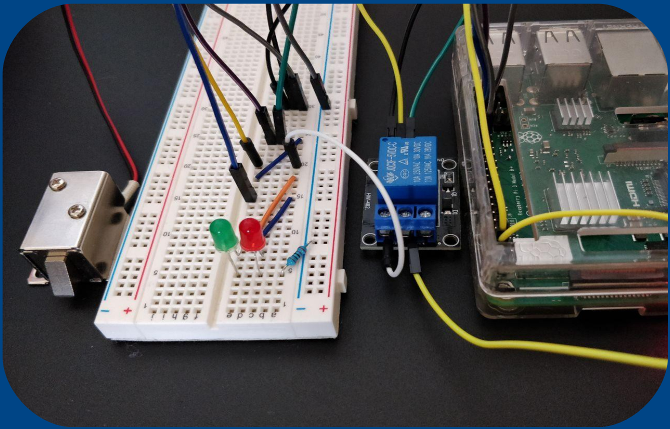
Use cases

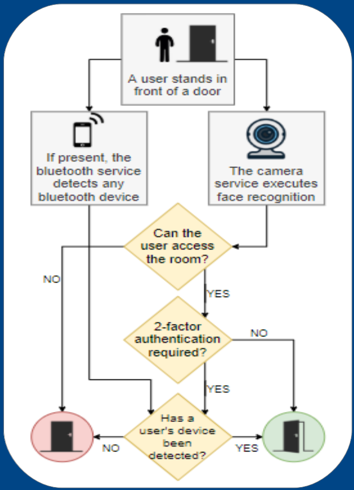
- ✓ Access control via face recognition in personal officies, departments or in private homes.
- ✓ Possibility to use a two-factor authentication in rooms that require more security.
- ✓ Thanks to its modularity, it is even possible to use our system in contexts where the lock is far from the camera, for example to control access in a car gate.

Prototype and demonstration

For logistical problems, we had to implement the lock service and the central service on one rasperry and the camera service and the bluetooth service on the other.

The demonstration will show the correct functioning of the system in all the possible cases.




A user stands in front of a door
If present, the bluetooth service detects any bluetooth device
The camera service executes face recognition
Can the user access the room?
NO
YES
2-factor authentication required?
NO
YES
Has a user's device been detected?
NO
YES

System architecture

We decided to implemented the system on two different Raspberry: one for the central service and one for the door service.

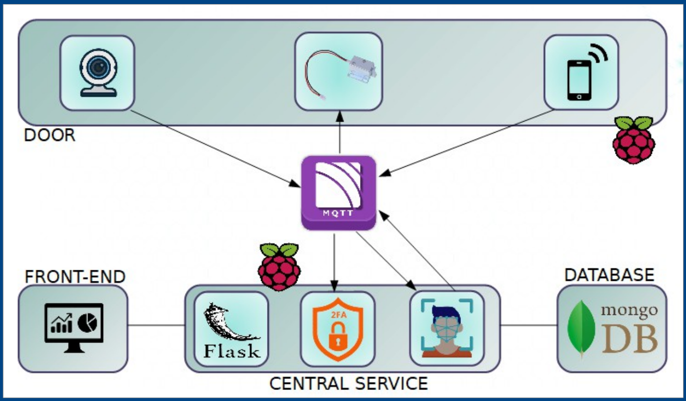
Central service:

- Database to store users, doors and policies data and the data related to the bluetooth devices position.
- Dashboard to add users, rooms or access policies.
- It receives access requests from the door service and sends responses based on database data.

Door service: it consists of three independent services:

- Lock service: it receives the responses from the central service and opens the lock
- Camera service: it deals with face recognition and sends access requests to the central service
- Bluetooth service (optional): it detects bluetooth devices near the door and sends their mac address to the central service.

→ Requests and responses are exchanged via mqtt.



Technologies

Raspberry PI 3 Model B+ e Raspberry Pi 3 Model B

Electronic lock:

- RPi.GPIO python library

Camera:

- face-recognition python library
- open-cv python library

Bluetooth:

- Pybluez python library

MQTT:

- Mosquitto as mqtt broker
- paho-mqtt for mqtt clients

Central service:

- Flask
- MongoDB Atlas

Future Works

- Extend the system, connecting more locks and cameras
- Bluetooth pairing instead of bluetooth mac comparison
- Add other authentication factors, such as NFC
- Integrate the system with user localization via bluetooth devices