Multifactor lock

fiveguys group

Quentin Beauchet

Yaacoub Yaacoub

Francesco Diana

Pietro Ventrella

Forner Yann

Content

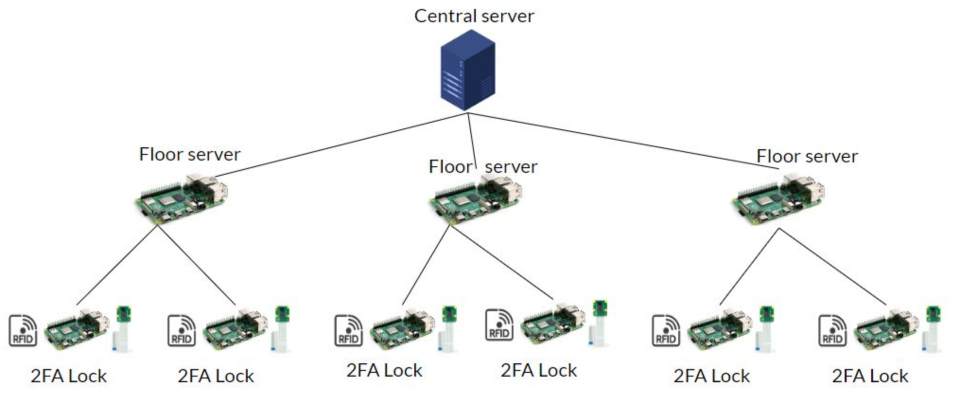
Introduction

- The purpose of the project is to design an electronic lock system with 2FA, working with facial recognition and RFID badge.
- The system is designed to work in an hypothetical scenario of a hotel or an office where there will be the need of different authorization levels for the rooms.
- We suppose to deploy our solution in an environment made of multiple floors or buildings, the system will be composed of a central server, several building or floor servers and a set of locks.

Proposed Solution

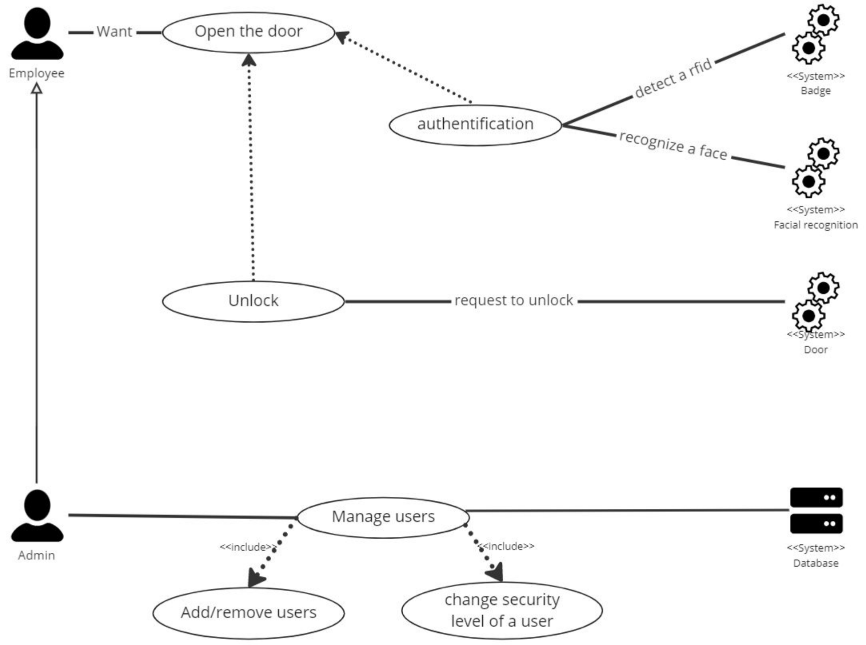
To unlock the door, the following steps have to be respected:

1. An employee / customer scans his RFID badge.
2. The Raspberry connected to the lock will send the id to the floor server.
3. The floor server will check the presence of the id in the database
4. Then, in case of a positive response, the Raspberry will take a picture of the user and eventually unlock the door.



Application architecture

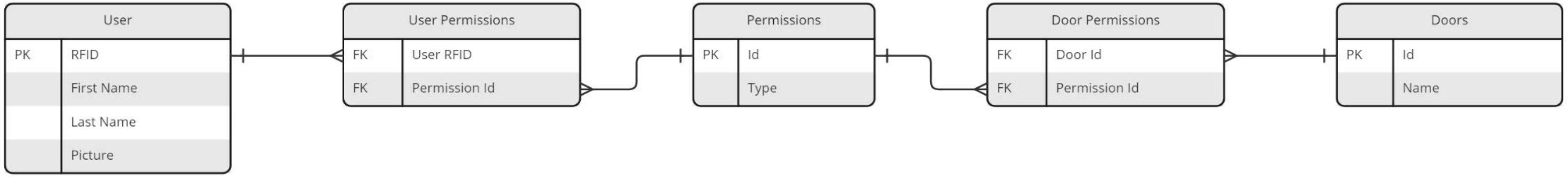
- The user will try to get access to a room scanning its badge.
- Then, the Raspberry will send, using a WiFi connection, the user id to the floor server with an HTTP GET request message.
- Each local server will be uniquely reachable through a endpoint.
- The connection between the floor servers and the central server will also use WiFi.
- Each local server will be uniquely reachable through a socket and it will exchange HTTP messages with the central server.
- The Raspberry and his lock communicate through Bluetooth



Application architecture

Then, concerning the application retrieval of data from the local server, we deploy for each floor server a MySQL database composed of the following tables:

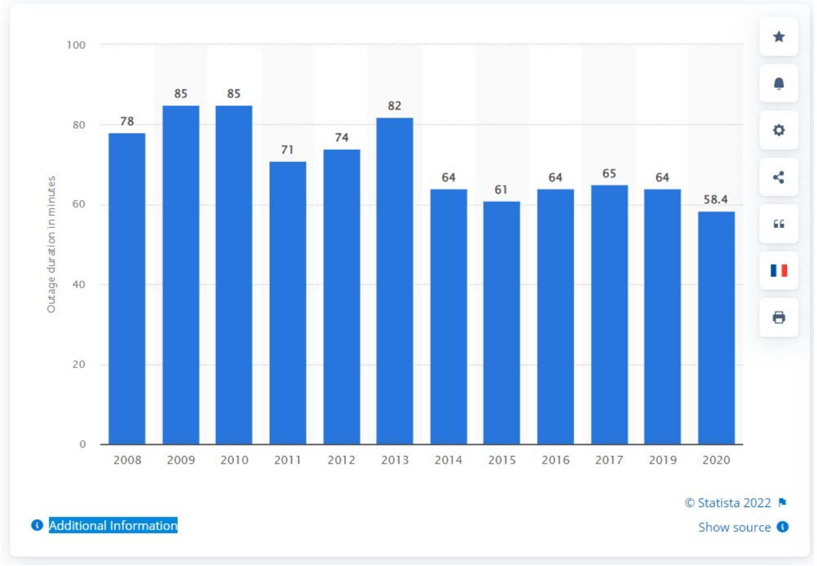
- permissions: this table contains the *id*, and the type of user related.
- doors: this table contains the information related to the *id* of a door and its *name*.
- user permission: here are stored the *user rfid*, used and a *permission id* which identifies the authorization level of the user.
- door permission: here are stored the *door id*, and a *permission id* which identifies the authorization level of the user.
- users: it contains information related to the users, including their *RFID* tag value, *first name*, *last name* and of course the *picture*, saved as a *blob* type.



Hardware components

- Each floor server will be provided with a 64GB SD card in order to store the user's information.
- In addition, each Pi connected to a lock and each floor server will be connected to an auxiliary battery.
- The estimated power consumption of a lock Pi are:
    - 900 mA for a Raspberry Pi 3 B+ in working mode.
    - 250 mA for the Raspberry Pi Camera Module 2.
    - 100 mA for the RFID Reader SCL3711.

- To power each lock for 8h it is needed, considering a tension of 5 V and an average current of 1250 mA , a power of 50W is needed which corresponds to a battery pack of capacity at least 10,000 mAh .



Estimated costs

- Moving on to an estimate of costs, considering the Raspberry connected to the lock, there are the following estimated prices:

- Raspberry Pi 3 B+ €35,00
- Pi Camera 2 € 25,00
- RFID Reader SCL371 € 30,00
- Micro SD 64 GB €10,00
- Power bank/battery €20,00

- Then, there is also to take into account the cost of the lock, that could vary based on the type of lock chosen. Instead, for what regards the servers, the estimated cost will be given by the Raspberry and the Micro SD card, so it will be around €45,00.

State of the practice

Nuki Smart Lock 3.0

Advantages:

- Ease of installation
  - Doesn't require an IT specialist
  - 3-minutes check for the door
- Smartphone interaction
  - Auto Unlock feature
  - Possibility to give authorization to guests
- Activity log

Disadvantages:

- Doesn't have 2FA
- Not scalable
- Need to change batteries

State of the practice

Orbitatech E4041 LCD Smart Electronic Hotel Lock

Advantages:

- LCD display to show informations
- Scalability
- Use of RFID tag or smartphone to unlock
- Integration in many areas

Disadvantages:

- Doesn’t have 2FA
- Considerable cost
- Battery life of 12 months

State of the practice

Openpath Video Reader Pro

Advantages:

- Security device
- Interoperability
- Use of RFID tag or smartphone to unlock
- Simple management

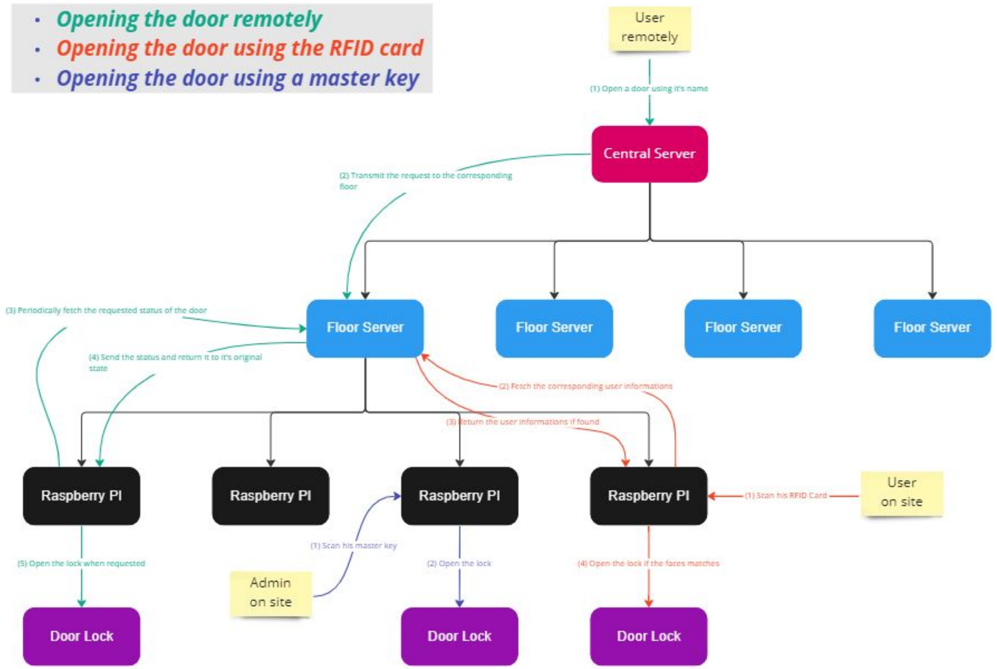
Disadvantages:

- Doesn't have 2FA
- Considerable cost

Example of usage

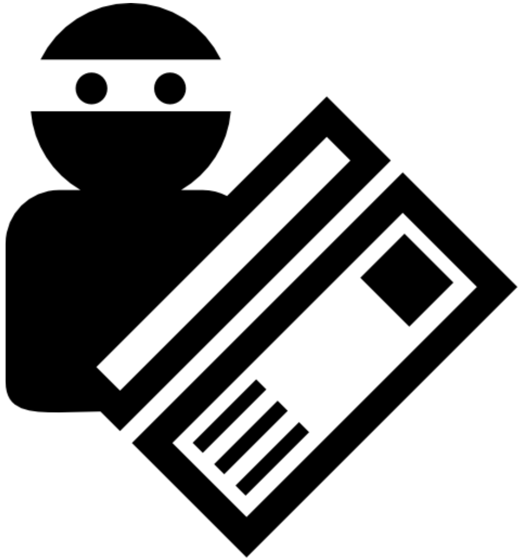
In the picture we described the 3 ways for an user to open a lock:

- Scanning the RFID card with the Raspberry connected to the lock.
- Using the RFID card owned by the admin.
- Remotely, using a dashboard of an application, even though for this demo we do it using the central server API routes directly.

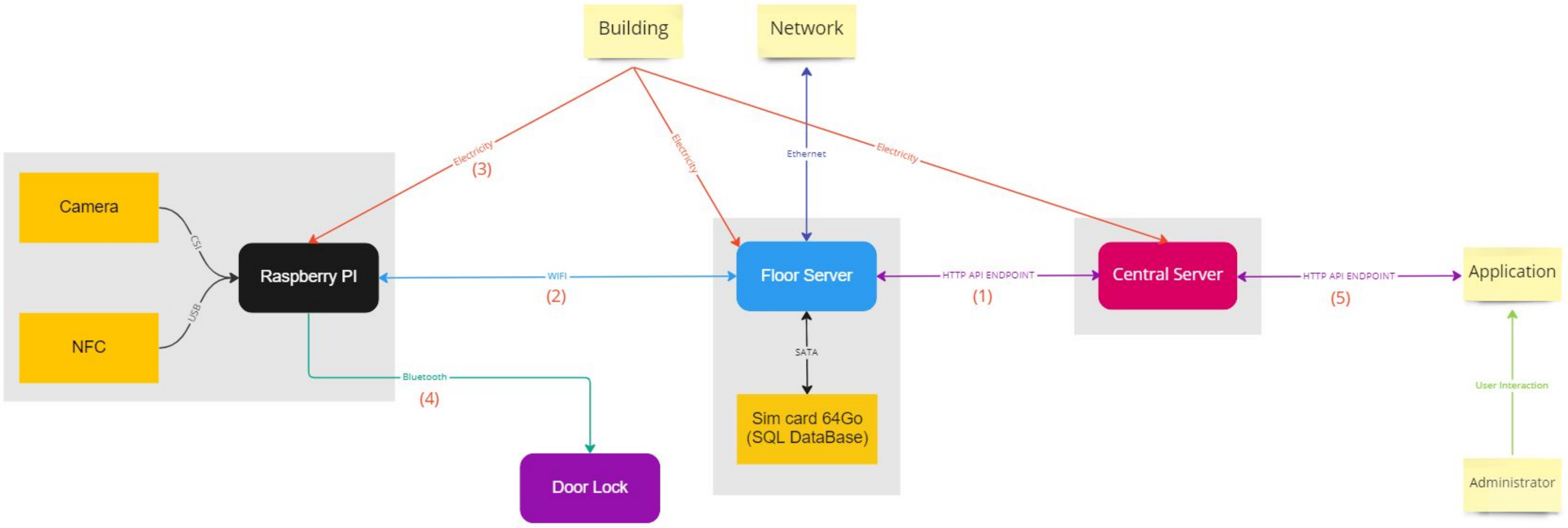


Opening the door remotely
Opening the door using the RFID card
Opening the door using a master key
User remotely
(1) Open a door using it's name
Central Server
(2) Transmit the request to the corresponding floor
Floor Server
Floor Server
Floor Server
Floor Server
(3) Periodically fetch the requested status of the door
(4) Send the status and return it to it's original state
(2) Fetch the corresponding user informations
(3) Return the user informations if found
Raspberry PI
Raspberry PI
Raspberry PI
Raspberry PI
(1) Scan his RFID Card
User on site
(5) Open the lock when requested
(1) Scan his master key
Admin on site
(2) Open the lock
(4) Open the lock if the faces matches
Door Lock
Door Lock
Door Lock

Critical analysis - Security concerns

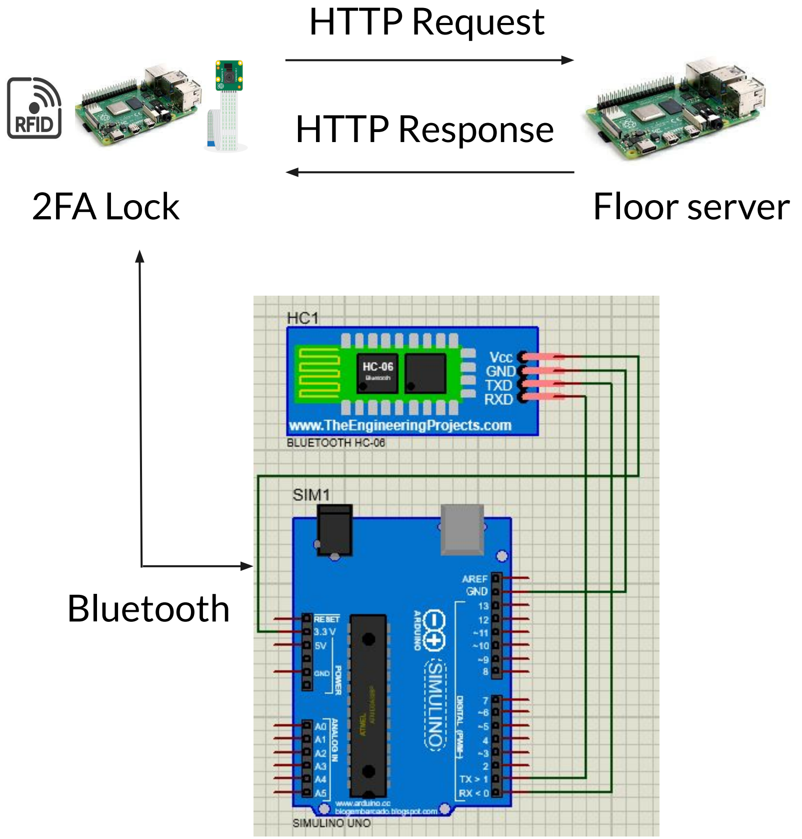


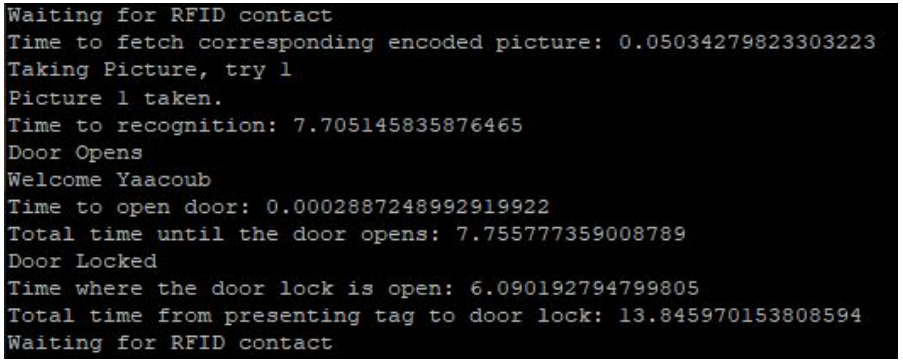
Critical analysis - Architecture weakness



Implementation - Door Side

Lock Control
- Communicates with the floor server with HTTP via Wi-Fi
- Read RFID, send it to server
- Receive encoded image from server in case we have a match
- Take a picture of the user
- Compare with fetched image
  - In case match
    - Send open signal to lock



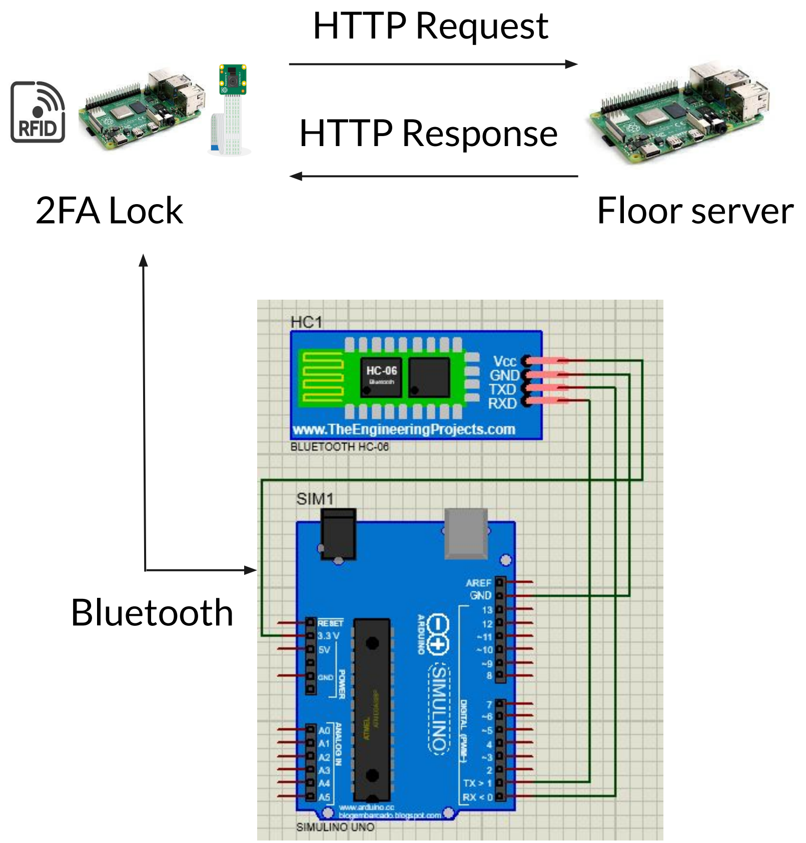


```
Waiting for RFID contact
Time to fetch corresponding encoded picture: 0.05034279823303223
Taking Picture, try 1
Picture 1 taken.
Time to recognition: 7.705145835876465
Door Opens
Welcome Yaacoub
Time to open door: 0.0002887248992919922
Total time until the door opens: 7.755777359008789
Door Locked
Time where the door lock is open: 6.090192794799805
Total time from presenting tag to door lock: 13.845970153808594
Waiting for RFID contact
```

Implementation - Door Side

Lock

- Simulated with an Arduino Uno and a LED
  - Door unlocked → LED On
  - Door Locked → LED Off
- Connected to the lock control with bluetooth
  - Built in Bluetooth module on the Raspberry Pi side
  - HC-06 bluetooth module on the arduino Uno side
- Lock receive a signal from door control to open
- Unlock and waits for 5 seconds
- Lock the door and send signal to inform the control that the door is locked.



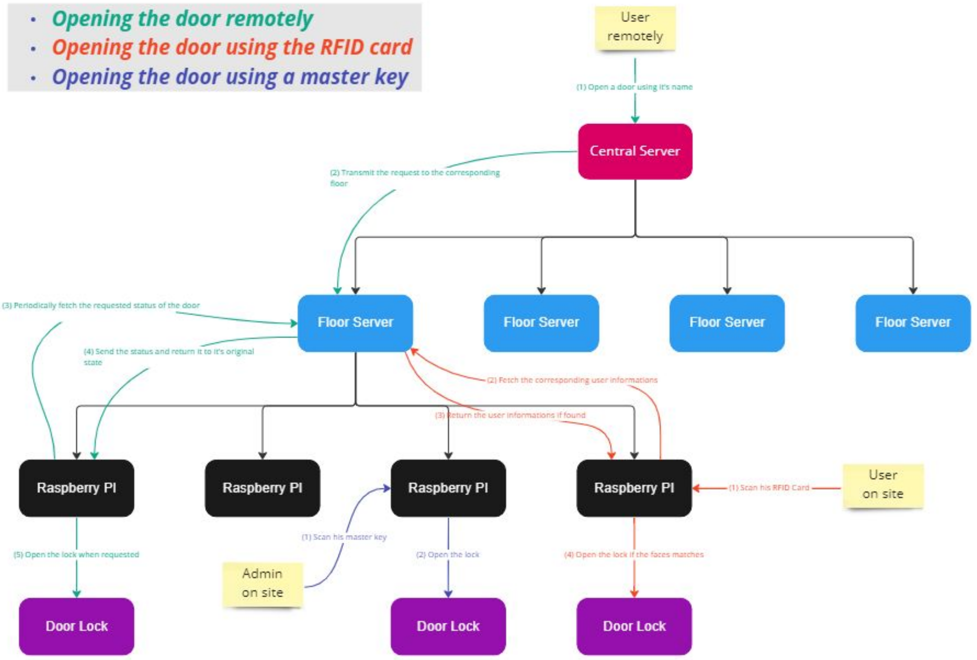
Server implementation

Central Server

```
> /api/update?door=`...`
```

Floor Server

```
> `GET`/api/open?door_id=`...`
> `GET` /api/status?door_id=`...`
> `GET` /api/user?door_id=`...`&rfid=`...`
> `GET` /api/admins
> `POST`
/api/user/add?rfid=`...`&first%20name=`...`
&last%20name=`...`&picture=`...`
```



Opening the door remotely
Opening the door using the RFID card
Opening the door using a master key
User remotely
(1) Open a door using it's name
Central Server
(2) Transmit the request to the corresponding floor
Floor Server
Floor Server
Floor Server
Floor Server
(3) Periodically fetch the requested status of the door
(4) Send the status and return it to it's original state
(2) Fetch the corresponding user informations
(3) Return the user informations if found
Raspberry PI
Raspberry PI
Raspberry PI
Raspberry PI
(1) Scan his RFID Card
User on site
(5) Open the lock when requested
(1) Scan his master key
Admin on site
(2) Open the lock
(4) Open the lock if the faces matches
Door Lock
Door Lock
Door Lock

Future implementations

- A dashboard or a mobile application used to manage, we already made a beta version of this tool as the cli but it’s only capable of adding users for now.
- Adding security to the communications and to the master key.
- Implement a spanning tree-like protocol that could dynamically reconnect Raspberry to another floor-server.
- Local cache in each lock.
- Change the architecture of the Raspberry so that we will able to do some micro-services for each auth factors.
- Use docker containers to update the Raspberry software more easily.
- Improve the face detection algorithm.
- Test if using a Raspberry 4 would improve the speed of the face recognition.

Conclusions

- Advantages of our solution:
  - scalability
  - 2FA
  - keyless
  - works with power outage

- Disadvantages:
  - need of an IT service
  - some weaknesses still to be countered
  - need of an admin