

# Conception d'un objet connecté



Louis Escamilla  
Benjamin Aupest  
Pierre-Manuel Lheureux  
Rémy Decourcelle - Emil Toulouse  
An-Vinh Lala - Valentin Chassignol  
Garice Morin - Thibault Tan  
Matthieu Baronnet - Ahmad Harkous  
Lise Pedemonte

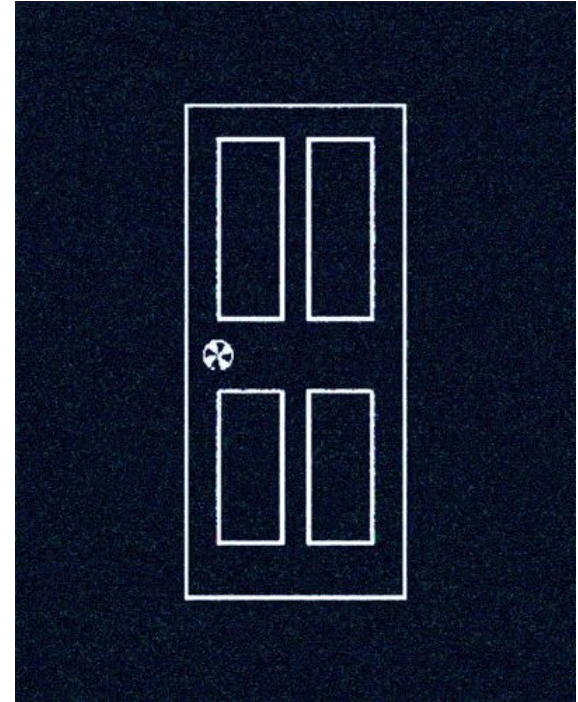
# Sommaire

- Présentation du sujet
- Conception et Fabrication
- Partie Software
- Partie Administrative

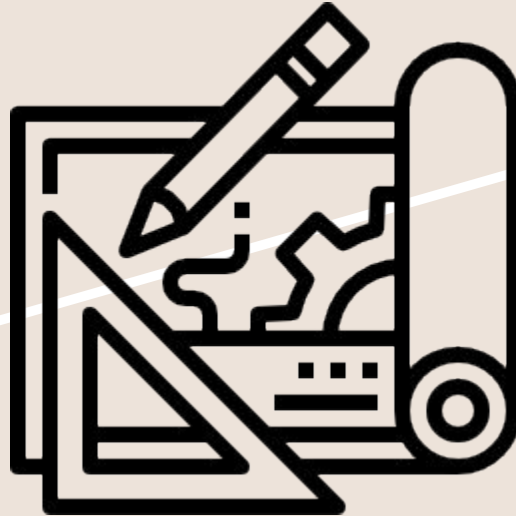
# Sujet

Conception d'une porte connectée

La porte s'ouvre grâce à un badge ou une carte (RFID)



# Partie Conception/Fabrication

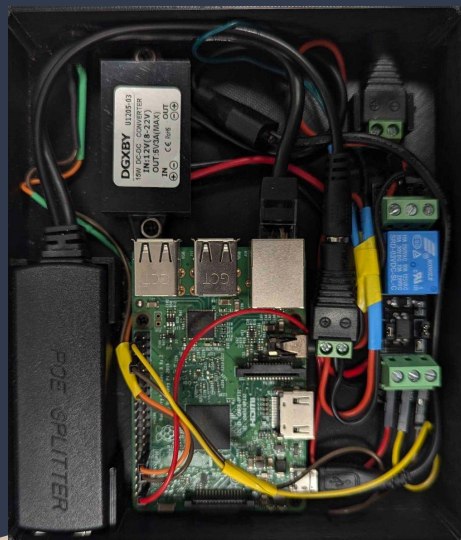


# Notre démarche de fabrication et de conception



- 1 - Proposer des idées et sélectionner la solution que l'on veut mettre en place
- 2 - Commander le matériel nécessaire
- 3 - Concevoir les pièces que l'on peut fabriquer nous-mêmes
- 4 - Fabriquer les pièces
- 5 - Assembler les pièces
- 6 - Ecrire les scripts pour gérer les signaux reçus
- 7 - Tester le bon fonctionnement du système

# Boitier interieur



## Objectifs :

Ce boîtier sert à **contrôler l'électro aimant**, il est situé à l'**intérieur** de la pièce sécurisée. Il est alimenté par **POE**. Après l'appui sur le **bouton** ou réception d'un ID de **carte/badge valide**, il déverrouille la porte. Il est connecté au boîtier extérieur par le **BLE**.

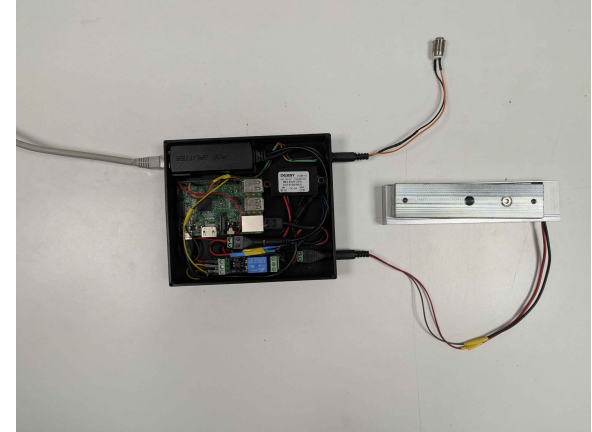
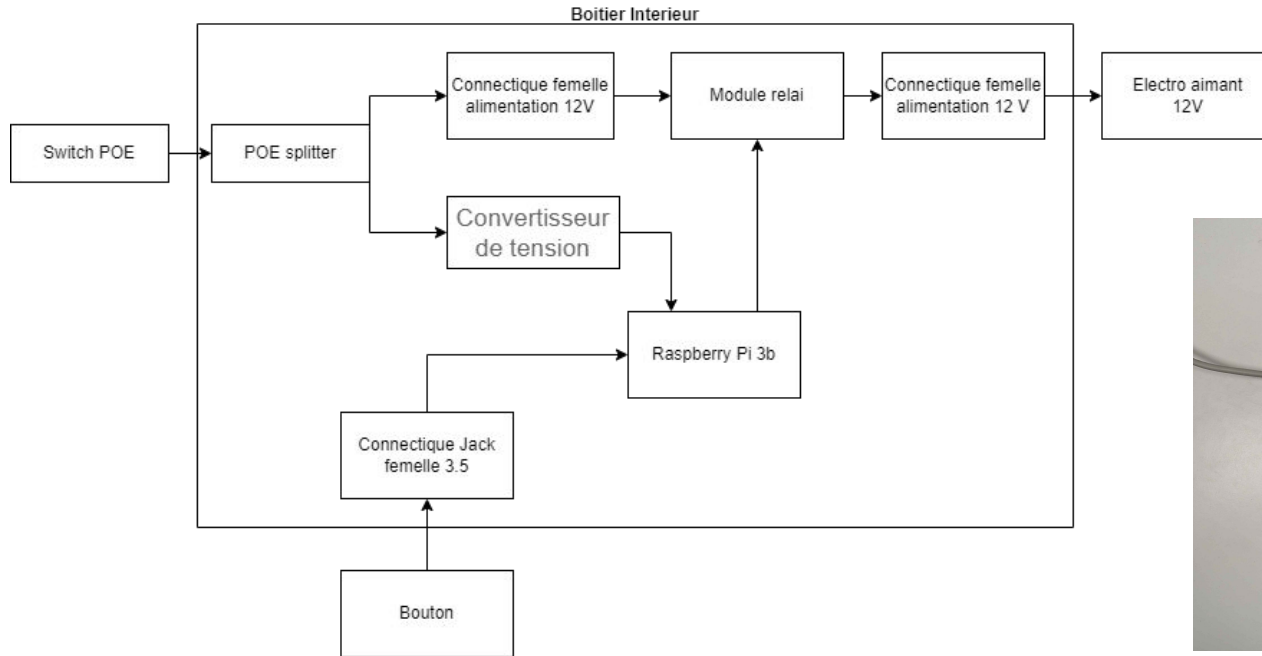
## Composants Internes :

- Raspberry Pi 3 Model B
- POE splitter
- Module relai
- Convertisseur de tension (12V -> 5V)
- Connectique femelle alimentation 12V
- Connectique Femelle JACK 3.5 (bouton externe)
- Connectique Femelle alimentation 12V (electro aimant)

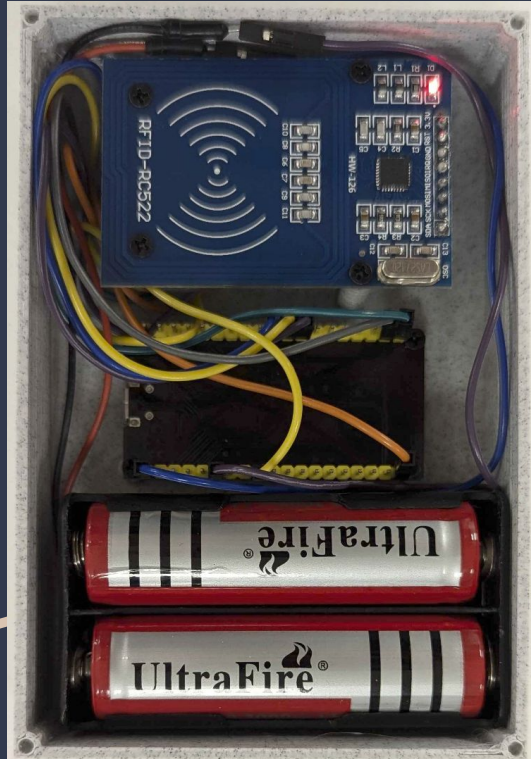
## Composants Externes :

- Electro aimant
- Switch POE
- Bouton d'ouverture intérieur

# Schéma d'architecture



# Boitier exterieur



Objectif :

Lire des badges/cartes RFID à l'extérieur de la pièce sécurisée et envoyer les informations au boitier intérieur en utilisant le BLE. Ce système fonctionne sur batterie.

Composants

- Module RFID
- ESP32
- Batteries 18650 3000mAh x2

Fonctionnement :

L'ESP est alimenté en 7.2V par les deux cellules 18650. Le code qui tourne dans l'ESP est un serveur bluetooth qui publie l'ID du badge/carte RFID scanné.

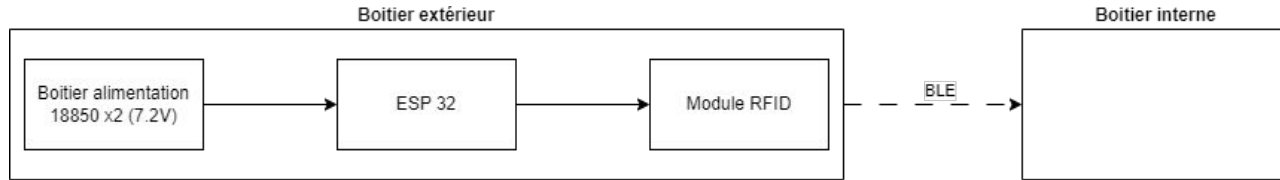
Autonomie du système : 7 mois de service.

Boitier :

Impression 3D FDM en PLA 0.2 mm 15% de remplissage.



# Schéma d'architecture



# Présentation (vidéo)



# Améliorations



## Boîtier intérieur :

- Cable management
- Utilisation d'un ESP32 pour gagner de la place

## Boîtier extérieur :

- Ajout d'un bouton reset
- Ajout d'une led pour visualiser si le badge est valide ou non
- Ajout d'un buzzer pour les mêmes raisons
- Ajouter un détrompeur à l'emplacement du module RFID.

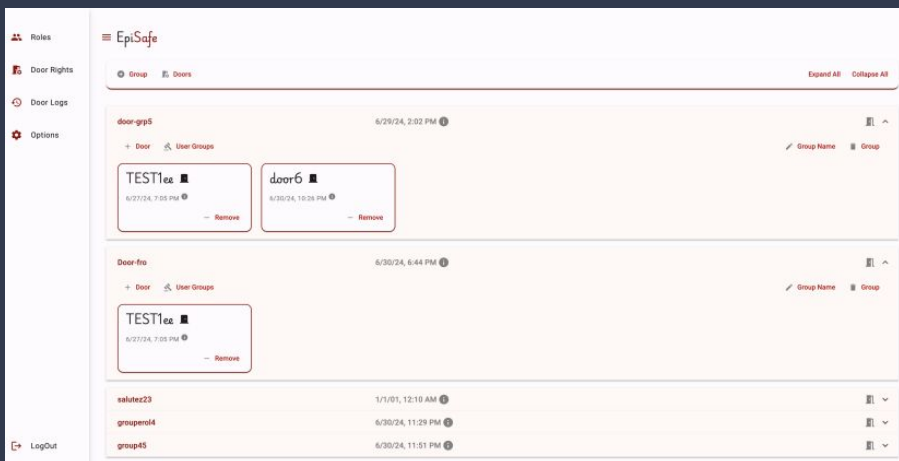
## Autres :

- Fabriquer un boîtier pour le bouton intérieur
- Gainer le câble du bouton intérieur

# Partie Software Serveur

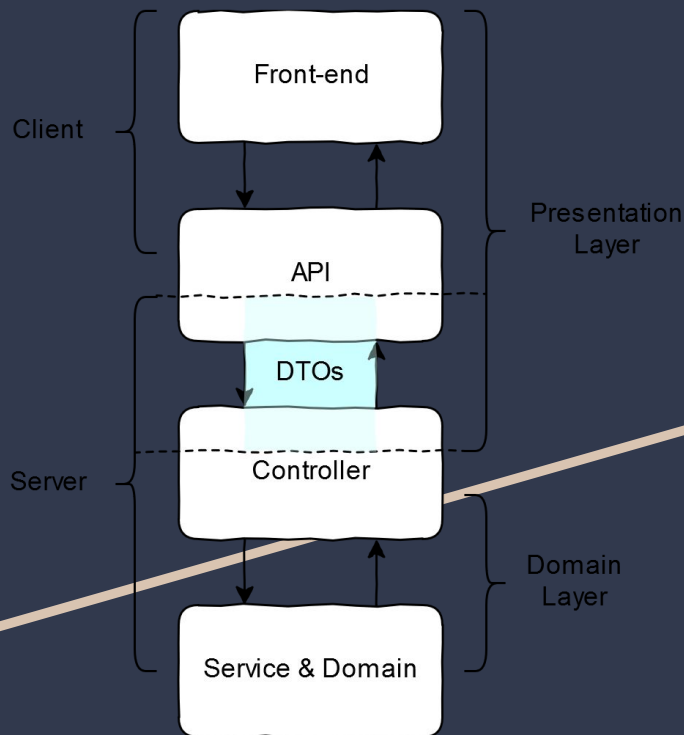


# Buts de notre software



- Ajouter/Supprimer des utilisateurs à nos groupes
- Accéder aux logs quand une porte s'ouvre, quand on ajoute/supprime un utilisateur/porte dans un groupe

# Backend



Composants:

- Une base de données Supabase

Backend en C# dont:

- 3 controllers pour les endpoints
- 4 dto afin de retourner le bon objet au Front
- 7 modèles pour chaque table dans Supabase
- Plusieurs services qui gèrent les exceptions et le getClient()

Connexion à Supabase via launchSettings dans le programme

# API C#

Episafe (API) v1.0.0 QA55  
<https://localhost:7232/swagger/v1/swagger.json>

Authorize

**Frontend** ^

- GET /admin/user/all ^
- GET /admin/user/{id\_user} ^
- DELETE /admin/user/{id\_user} ^
- POST /admin/user ^
- PATCH /admin/user ^
- GET /admin/usergroup/all ^

**Logs** ^

- GET /logs/all ^
- POST /logs/delete ^

**Open** ^

- POST /open/door ^

3 API: API Login pour la gestion de l'ouverture des portes, API Logs et API Users pour la gestion des utilisateurs/portes/groupe.

8 tables:

- 4 tables gestion des portes/users
- 3 tables de jointure
- 1 table logs

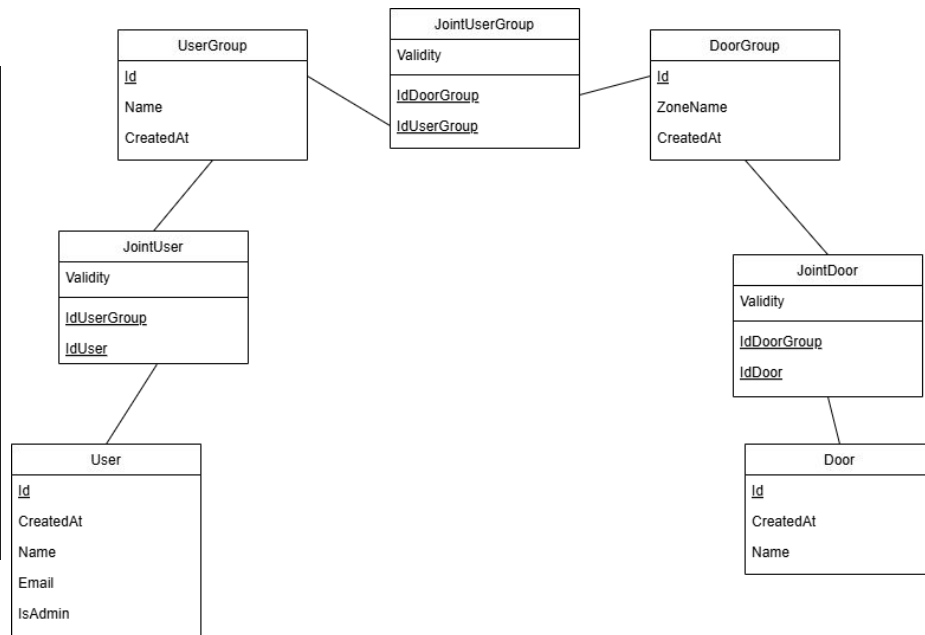
Endpoints: CRUD pour les 4 tables, getALL/suppAll pour les logs et Post/Delete pour les affectations aux groupes

# Supabase et architecture

Database Tables > users

< 🔍 Filter columns + New column

Name	Description	Data Type	Format	
id	No description	bigint	int8	⋮
created_at	No description	timestamp with time zone	timestampz	⋮
name	No description	character varying	varchar	⋮
email	No description	character varying	varchar	⋮
is_admin	No description	boolean	bool	⋮





# Frontend

- Google Material Design 3.0
- Angular V18



Angular Material

AG Grid 

# (Démo)

Roles

Door Rights

Logs

LogOut

EpiSafe

Group Users

Expand All Collapse All

groupe1

6/29/24, 9:46 PM

groupe1

+ User

Group Name Group

ahmad

ahmad.harkous@epita.fr

6/30/24, 2:42 PM

Remove

user3

user@epita.fr

6/30/24, 11:28 PM

Remove

groupe2

6/29/24, 8:57 PM

groupe2

groupe3

6/30/24, 5:20 PM

groupe3

group4

6/30/24, 11:28 PM

group4

demo1

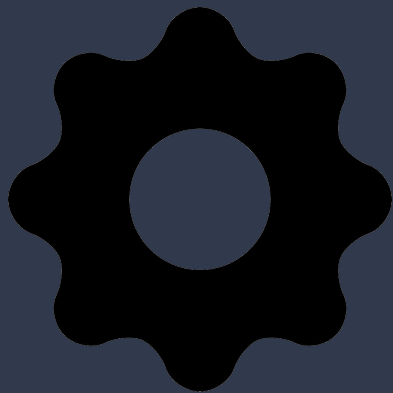
7/16/24, 8:55 PM

demo1

© Harkous Ahmad 2025, Google M3

18

# Améliorations



- Renforcer le login avec un keycloak pour l'authentification
- Effectuer l'étape de déploiement
- Améliorer le système d'ouverture actuel sur le backend et faire le lien avec le RFID de la serrure
- Ajouter le système de lockdown
- Ajout d'un QR code d'ouverture temporaire, uniquement via le réseau interne.

# Partie Administrative



# Processus et industrialisation



Création d'une fiche processus, matériel et coût.

Analyse et optimisation des coûts pour une seconde version.

# Service après vente



Création d'une fiche de réparabilité et  
calcul de l'indice de réparabilité.

Création d'une fiche produit (qualité,  
information générale ...)

# Merci pour votre écoute

Nous serons heureux de répondre à vos questions