

INGINIERIE INFORMATIQUE ET SYSTEMES EMBARQUES

CAHIER DE CHARGE

Effectué du : 01/04/2025

CONTROLE D'ACCES A UNE PORTE VIA UNE APPLICATION MOBILE

Réalisé par :


- SARMA FATIMA ZAHRA
- MAICHNI HAJAR
- SEBBAR MOHAMED
- SAGO YOUNES

Encadrant :

- Prof. JAAFAR IDRAIS

Année Universitaire

2024/2025



Tables de contenu:

I.	INTRODUCTION:	4
II.	Architecture logicielle :	5
1.	Description générale de l'architecture du système	5
i.	Composants principaux :	5
ii.	Fonctionnement Général du Système :	5
iii.	Technologies Utilisées :	6
2.	Déploiement Distribué :	6
i.	Topologie du réseau :	6
ii.	Sécurité et Sessions :	7
III.	Diagrammes UML	7
1.	Diagramme des Cas d'Utilisation :	7
2.	Diagramme de Classes :	8
3.	Diagramme de Séquence :	8
4.	Diagramme de Déploiement :	9
5.	Diagramme de Composants :	10
IV.	Tests et Validations :	11
1.	Tests Unitaires :	11
2.	Tests d'Intégration :	12
3.	Tests Réels (Simulés dans Wokwi) :	13
4.	Perspectives d'Amélioration :	13

Tables de figures

Figure 1: Diagramme de cas d'utilisation	8
Figure 2: Diagramme de classes.	8
Figure 3: Diagramme de séquence.....	9
Figure 4: Diagramme de déploiement.....	10
Figure 5: Diagramme de composants.	11
Table 1: technologies utilisées	6

I. INTRODUCTION:

Dans un monde où la sécurité des biens et des personnes est devenue un enjeu majeur, le contrôle d'accès physique représente un domaine stratégique. Les systèmes traditionnels (clés mécaniques, badges passifs, verrous manuels) sont aujourd'hui confrontés à leurs propres limites : absence de traçabilité, impossibilité de supervision à distance, vulnérabilité aux duplications ou pertes de clés, et manque de réactivité en cas d'intrusion.

Parallèlement, la connectivité croissante des objets (IoT), la démocratisation des réseaux sans fil, et la généralisation des smartphones offrent de nouvelles perspectives pour réinventer les mécanismes de sécurité. Les utilisateurs cherchent désormais des solutions intelligentes, flexibles et accessibles à distance, capables non seulement de gérer les accès en temps réel, mais aussi de les superviser, les historiser et de réagir aux incidents de manière automatisée.

C'est dans ce contexte que s'inscrit notre projet, dont l'objectif est de développer un système distribué de contrôle d'accès à une porte via une application mobile, associé à un lecteur RFID connecté. Plus précisément, le système permet :

- à un utilisateur autorisé d'accéder à la porte en scannant une carte RFID,
- au serveur central de vérifier, valider ou refuser l'accès,
- au propriétaire de consulter en temps réel tous les événements depuis une application mobile dédiée, et d'être notifié immédiatement en cas de tentative non autorisée.

Le projet repose sur une architecture distribuée, dans laquelle plusieurs composants — un microcontrôleur ESP32 connecté en WiFi, un serveur gRPC écrit en Python, et une application mobile — communiquent à distance pour accomplir ces fonctions de sécurité.

Problématique :

Comment concevoir une solution de contrôle d'accès basée sur RFID, fiable et distribuée, permettant une gestion centralisée, une supervision en temps réel, et une réaction immédiate en cas de tentative d'accès non autorisé, tout en garantissant une communication rapide, sécurisée et évolutive entre les composants du système ?

Ce projet vise à répondre à cette problématique en exploitant les technologies modernes de l'IoT et du développement mobile, tout en s'appuyant sur une architecture logicielle modulaire, sécurisée et performante, adaptée aux environnements domestiques ou professionnels.

II. Architecture logicielle :

1. Description générale de l'architecture du système

Le projet « Contrôle d'accès à une porte via une application mobile » repose sur une architecture distribuée intelligente permettant à un propriétaire de maison de contrôler l'accès à sa porte d'entrée à l'aide d'un système RFID connecté via WiFi à une plateforme de gestion centralisée.

Le système permet aux membres de la famille (préalablement autorisés) d'ouvrir la porte en scannant leur carte RFID, tandis que le propriétaire peut visualiser tous les accès depuis son application mobile. Toute tentative non autorisée est également enregistrée et peut être signalée.

Ce système utilise gRPC (Google Remote Procedure Call) comme mécanisme de communication distante, offrant des performances élevées, une faible latence, et une sécurité accrue dans un environnement distribué.

i. Composants principaux :

- ESP32 + lecteur RFID: Lit les cartes RFID et envoie l'UID via WiFi.
- Serveur gRPC (Python) : Authentifie les cartes, prend des décisions d'accès, envoie les commandes à l'ESP32 et journalise tous les événements.
- Application mobile du propriétaire (Flutter) : Permet au propriétaire de consulter les accès, ajouter/supprimer des cartes autorisées, et recevoir des notifications d'accès refusé.

ii. Fonctionnement Général du Système :

1. Un membre de la famille scanne sa carte RFID sur le lecteur connecté à l'ESP32.
2. L'ESP32 envoie l'UID de la carte au serveur via WiFi (par gRPC ou requête HTTP).
3. Le serveur vérifie si l'UID est autorisé :
 - Si oui : il envoie un signal d'ouverture à l'ESP32 et enregistre l'accès.

- Si non : l'accès est refusé, l'événement est enregistré, et une alerte peut être envoyée au propriétaire.

4. Le propriétaire peut, via l'application mobile :

- Gérer les cartes autorisées.
- Consulter les journaux d'accès.
- Être alerté des tentatives non autorisées.

iii. Technologies Utilisées :

Composant	Technologies
ESP32 + RFID	Simulé via Wokwi avec WiFi + lecteur RC522
Communication	gRPC (Python) ou HTTP (via WiFiClient)
Backend	Python (serveur gRPC avec vérification des droits)
Application mobile	Flutter
Protocoles	gRPC pour communication distante, HTTP/2, WiFi pour transmission

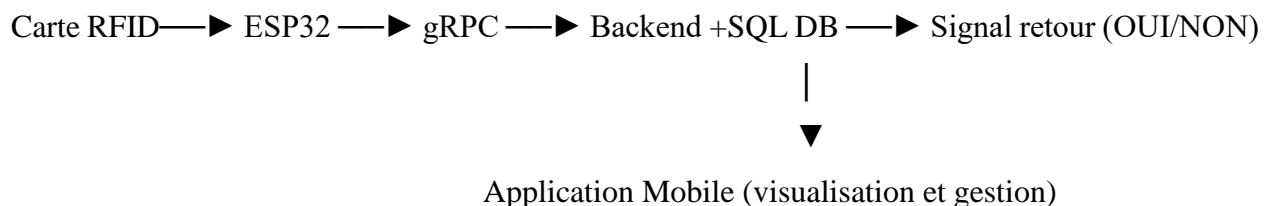
Table 1: technologies utilisées

2. Déploiement Distribué :

Le système est réparti entre plusieurs entités qui communiquent entre elles via le réseau :

- ESP32 + RFID : déployé sur un microcontrôleur simulé (Wokwi), connecté au WiFi.
- Serveur gRPC : hébergé sur une machine locale ou cloud. Il est l'unité de traitement central.
- Application mobile du propriétaire : installée sur son smartphone, utilisée pour la supervision.

i. Topologie du réseau :



ii. Sécurité et Sessions :

- Chaque carte RFID est liée à un utilisateur spécifique dans la base de données.
- Les tentatives d'accès sont journalisées avec :
 - ✓ UID de la carte
 - ✓ Heure et date
 - ✓ Statut (autorise / refuse)
- L'application mobile utilise une authentification propriétaire.

III. Diagrammes UML

1. Diagramme des Cas d'Utilisation :

Diagramme des Cas d'Utilisation

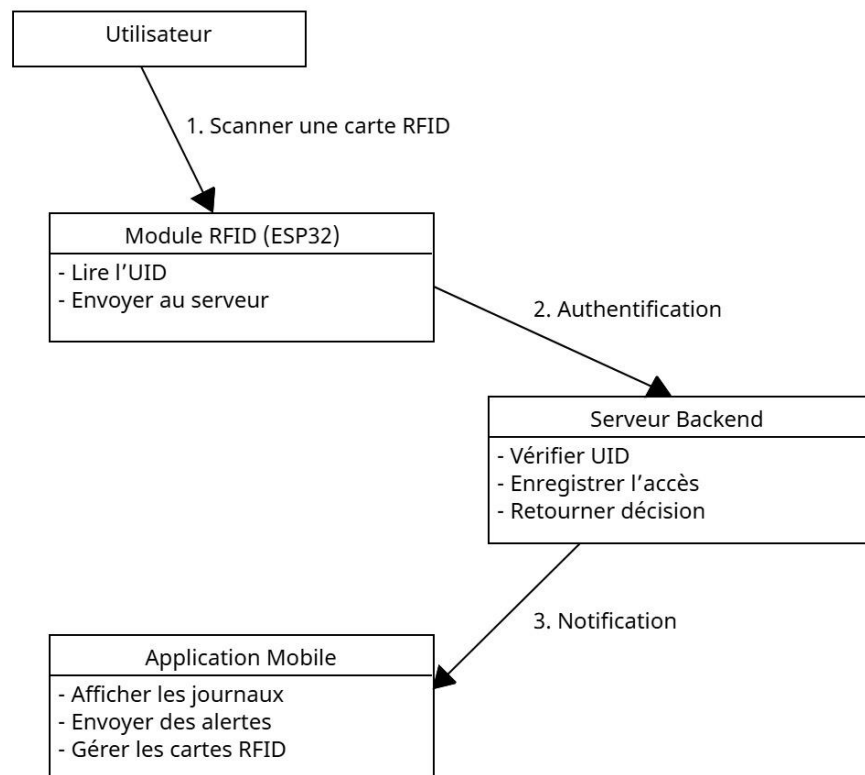


Figure 1: Ce diagramme décrit les principales interactions entre les utilisateurs et le système. Il inclut des cas d'utilisation comme l'Ajout d'une carte autorisée, la Consultation des logs d'accès, et la Réception de notifications sur l'application mobile. Les acteurs, comme l'Utilisateur ou l'Administrateur, interagissent avec le système pour gérer les accès à la porte, configurer des cartes autorisées, et recevoir des informations en temps réel.

2. Diagramme de Classes :

Diagramme de Classes

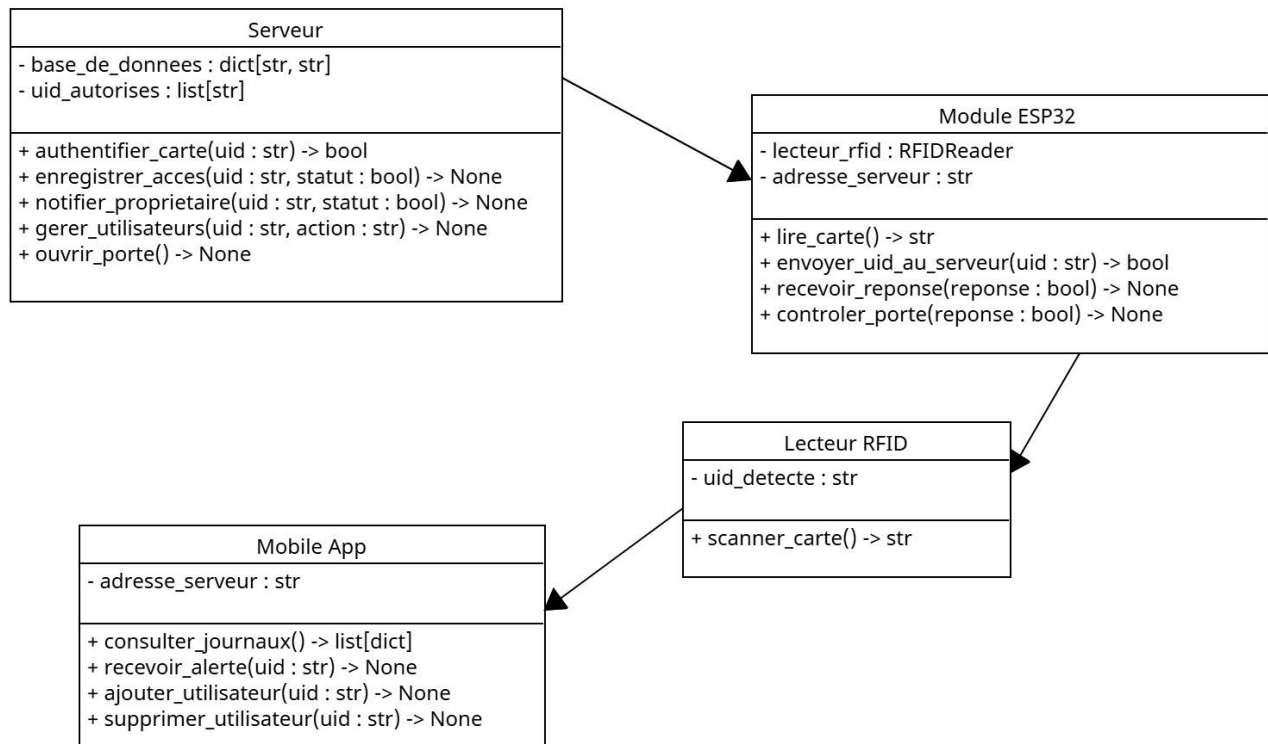


Figure 2: Ce diagramme définit la structure détaillée des classes du système, ainsi que leurs attributs et méthodes. Il inclut les classes utilisées. Chaque classe possède des variables et des méthodes qui définissent le comportement du système. Le diagramme montre également les relations entre les classes, telles que l'envoi de commandes entre le Serveur et l'ESP32, ou l'ajout de cartes autorisées dans la Base de données du Serveur.

3. Diagramme de Séquence :

Diagramme de Séquence

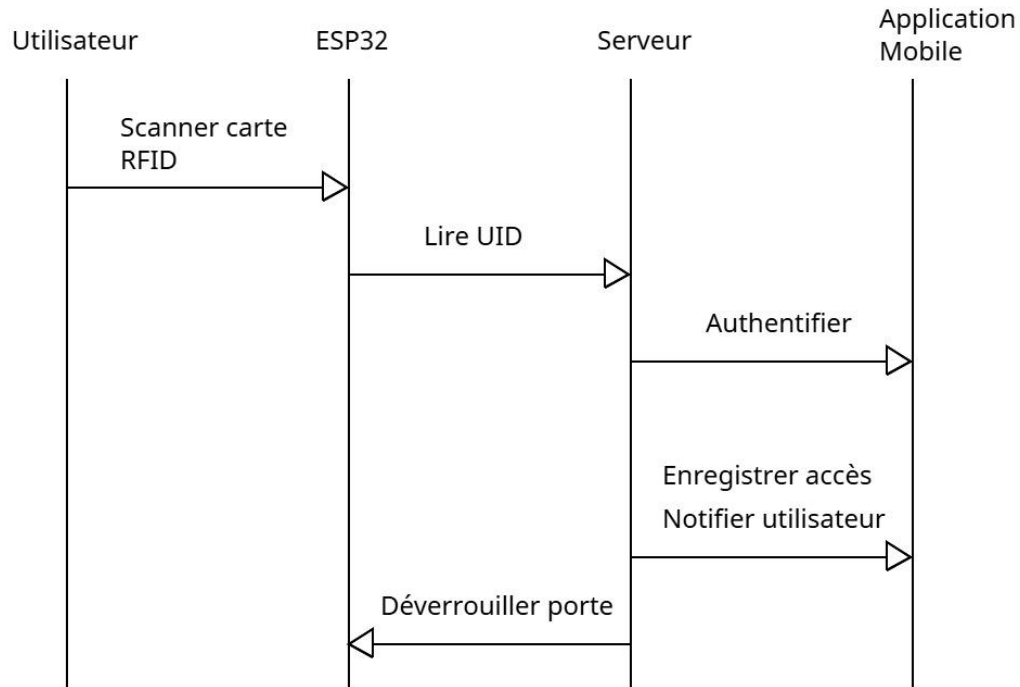


Figure 3: Ce diagramme montre l'ordre chronologique des interactions entre les composants du système lors de l'exécution d'une fonctionnalité spécifique. Il illustre comment les objets (comme ApplicationMobile, Serveur, et ESP32) échangent des messages pour accomplir une tâche. Par exemple, lorsqu'un membre de la famille scanne une carte RFID, l'ApplicationMobile envoie une requête au Serveur pour vérifier l'UID, et le Serveur envoie ensuite une commande au ESP32 pour ouvrir la porte, ou renvoie un message de refus si l'UID est non autorisé. Ce diagramme aide à visualiser le flux d'exécution et l'interdépendance entre les composants du système au cours de l'interaction.

4. Diagramme de Déploiement :

Diagramme de Déploiement

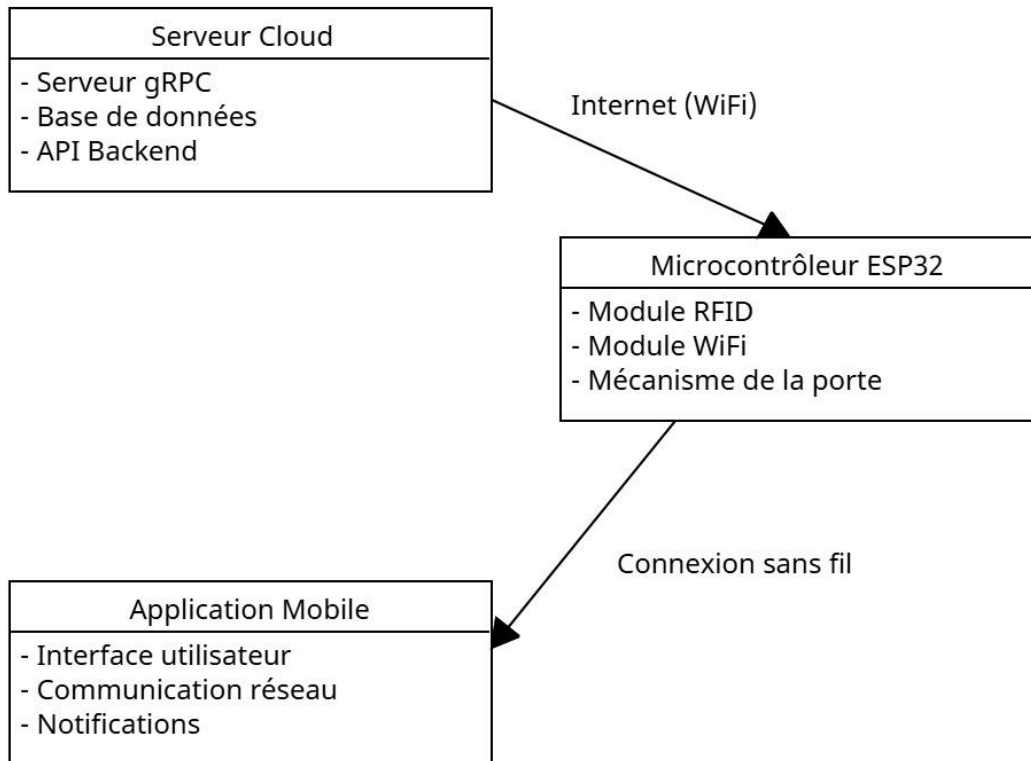


Figure 4: Ce diagramme montre comment les composants du système sont déployés dans l'environnement physique. Il inclut la répartition des serveurs, des ESP32, et des appareils mobiles sur le réseau. Il illustre comment ces composants communiquent entre eux à travers un réseau, que ce soit localement ou via le cloud, et comment chaque appareil (Serveur, ESP32, Application Mobile) est connecté pour assurer le bon fonctionnement du système.

5. Diagramme de Composants :

Diagramme de Composants

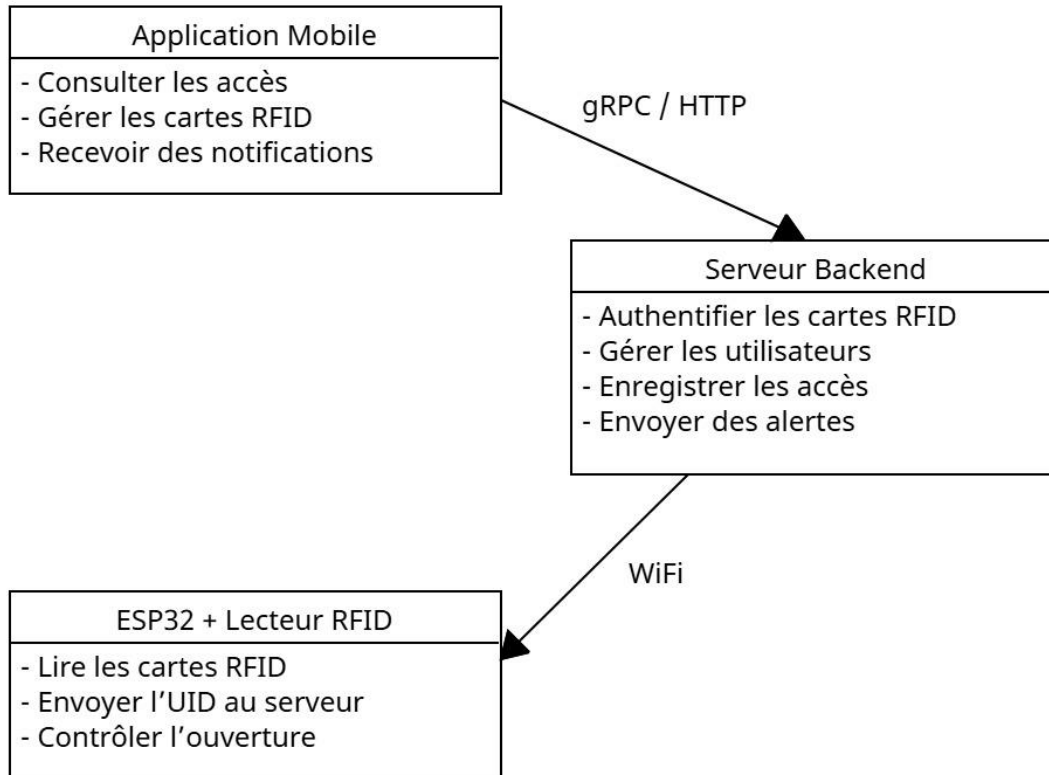


Figure 5: Ce diagramme représente les composants principaux du système et leur interaction. Il inclut des éléments comme le Serveur, l'ESP32, et l'Application Mobile, illustrant comment ils communiquent entre eux. Le Serveur vérifie l'accès des cartes RFID et envoie des commandes au ESP32 pour ouvrir ou fermer la porte. L'Application Mobile permet à l'utilisateur de gérer l'accès et de recevoir des notifications.

IV. Tests et Validations :

1. Tests Unitaires :

ESP32 (environnement simulé) :

- Simulation de lecture d'une carte RFID valide et invalide.
- Réactions attendues du microcontrôleur selon les signaux reçus (ouvrir / refuser l'accès).

Backend (gRPC) :

- Vérification de la validité des identifiants de carte RFID.
- Tests des fonctions de vérification des accès : retour « autorisé » ou « refusé » selon les droits.
- Tests des fonctions de journalisation : insertion correcte des logs dans la base de données.

Application mobile :

- Vérification de l'authentification du propriétaire.
- Tests des interfaces de consultation des accès.
- Ajout/suppression de cartes autorisées (test des entrées et validations de formulaire).

2. Tests d'Intégration :

Test complet du processus : lecture de carte → communication → décision du serveur → retour au microcontrôleur .

Vérification du flux d'informations entre :

- ✓ ESP32 ↔ Backend gRPC
- ✓ Backend gRPC ↔ Application mobile

Simulation de plusieurs scénarios :

- ✓ Accès autorisé
- ✓ Accès refusé
- ✓ Perte de connexion réseau
- ✓ Carte inconnue

3. Tests Réels (Simulés dans Wokwi) :

La simulation de l'environnement matériel a été réalisée à l'aide de Wokwi, en intégrant un ESP32 avec un lecteur RFID. Plusieurs UUIDs ont été simulés pour représenter différents utilisateurs (autorisés ou non). Cette simulation a permis d'observer le comportement du système face à diverses situations, notamment les délais de traitement, la réactivité de l'ouverture de la porte, ainsi que la cohérence des journaux d'accès générés.

4. Perspectives d'Amélioration :

- Reconnaissance faciale locale ou cloud pour les utilisateurs réguliers.
- Version Web de l'interface propriétaire en complément de l'application mobile.
- Ajout d'une caméra connectée à l'ESP32 pour capturer une photo lors de chaque tentative d'accès.