

Proposition de projet ZZ3 : Système de Gestion Intelligente pour une Maison Connectée

Nom : KHEDHAOURIA

Prénom : Eliès

ZZ3 F5 (Sécurité et réseau)

Description :

Développer une application avec QT pour gérer une maison intelligente connectée via un réseau local ou Internet. L'application permettra de superviser et contrôler divers appareils IoT comme les lumières, thermostats, caméras de surveillance, serrures intelligentes, etc., à distance et en temps réel.

Fonctionnalités proposées :

Interface utilisateur QT :

Créer une interface graphique intuitive qui permet de contrôler et surveiller plusieurs dispositifs dans la maison.

Interface dynamique qui affiche les données en temps réel (température, caméra en direct, statut des portes, etc.).

Système d'alarme.

Interface utilisateur via une application module (si le temps le permet) :

Créer une application mobile complémentaire pour que l'utilisateur puisse contrôler sa maison via son smartphone, en utilisant la même API réseau que l'application de bureau.

Communication Réseau

Inclure divers protocoles de communication avec les appareils IOT :

- **MQTT** et **Websocket** seront utilisés pour la communication en temps réel, permettant d'envoyer et recevoir des données instantanément (par exemple, allumer une lumière ou recevoir une alerte de sécurité).

- **HTTP** via une **API REST** sera utilisé pour les communications nécessitant des transactions plus standards, telles que la gestion des utilisateurs, l'enregistrement des appareils...

Sécurité Réseau :

- Implémenter un système de sécurité robuste pour authentifier les utilisateurs et **chiffrer** les communications entre l'application et les appareils connectés.

- L'utilisation de **SSL/TLS** au sein des divers protocoles afin de sécuriser les communications, et d'éviter toute interception malveillante.

Base de données locale/cloud :

- Stocker les données des capteurs et les événements dans une base de données locale ou dans le cloud pour avoir un historique des actions effectuées dans la maison (journal d'activité) (⇒ surveillance accrue par l'utilisateur).

- Pour une plus grande évolutivité et une meilleure gestion des données sur le long terme, l'application pourra être reliée à une base de données dans le **cloud** (par exemple, **Firebase** ou un serveur **MySQL**).

- Pour les données temporelles (par exemple, les variations de température ou les logs d'activité en continu), il est possible d'explorer l'utilisation de bases de données optimisées pour les séries temporelles, comme **InfluxDB** (j'avais utilisé cette technologie pendant mon stage).

Contrôle et automatisation :

- Développer des fonctionnalités d'automatisation : par exemple, créer des **scénarios personnalisés** (comme éteindre toutes les lumières à 23h ou activer les caméras quand l'utilisateur quitte la maison).

- Support pour des **règles basées sur des événements** : si la température dépasse 25°C, baisser automatiquement les stores.

Pour les tests :

- En l'absence de vrais appareils IoT, des **simulateurs d'appareils IoT** seront utilisés pour tester l'application :

- Par exemple, des **simulateurs MQTT** permettent de tester les interactions en temps réel avec des appareils simulés.
- Nous pourrions également développer nos propres simulateurs à l'aide de **programmes Python** ou utiliser des outils comme **Postman** pour tester les communications HTTP.
- Des plateformes en ligne comme **IoTIFY** ou **ThingSim** permettront de simuler des appareils virtuels via les protocoles **MQTT** et **HTTP**. Ces outils permettent de créer des dispositifs virtuels et de programmer leur comportement, reproduisant ainsi des environnements IoT réalistes.

- Si le temps le permet, il serait possible de créer une maquette physique utilisant des microcontrôleurs (comme un **Arduino**), un module Wi-Fi, et divers composants électroniques (capteurs de température, relais pour contrôler des lumières, etc.) pour tester directement sur des objets physiques.

Quelques points concernant ce projet :

Ce projet représente bien plus qu'un simple exercice technique. Il constitue pour moi l'opportunité de concrétiser une idée que j'ai toujours souhaité réaliser. La **domotique** est un domaine en constante évolution et particulièrement passionnant. De plus il s'agit d'utiliser des connaissances

théoriques acquises durant ces deux dernières années, que ce soit en sécurité ou en cryptographie à des fins concrètes.

En plus de me permettre de renforcer nos compétences en **développement logiciel** (notamment en **C++** et **Qt**), ce projet nous permettra d'améliorer nos connaissances en **réseaux** et en **sécurité informatique**.

Je souhaite aussi utiliser au plus des outils permettant la gestion de projet en groupe tels que Git (que je ne maîtrise pas de manière suffisante actuellement). La compétence 'Savoir utiliser Git' est j'ai remarqué demandé dans la majorité des offres d'emplois IT.

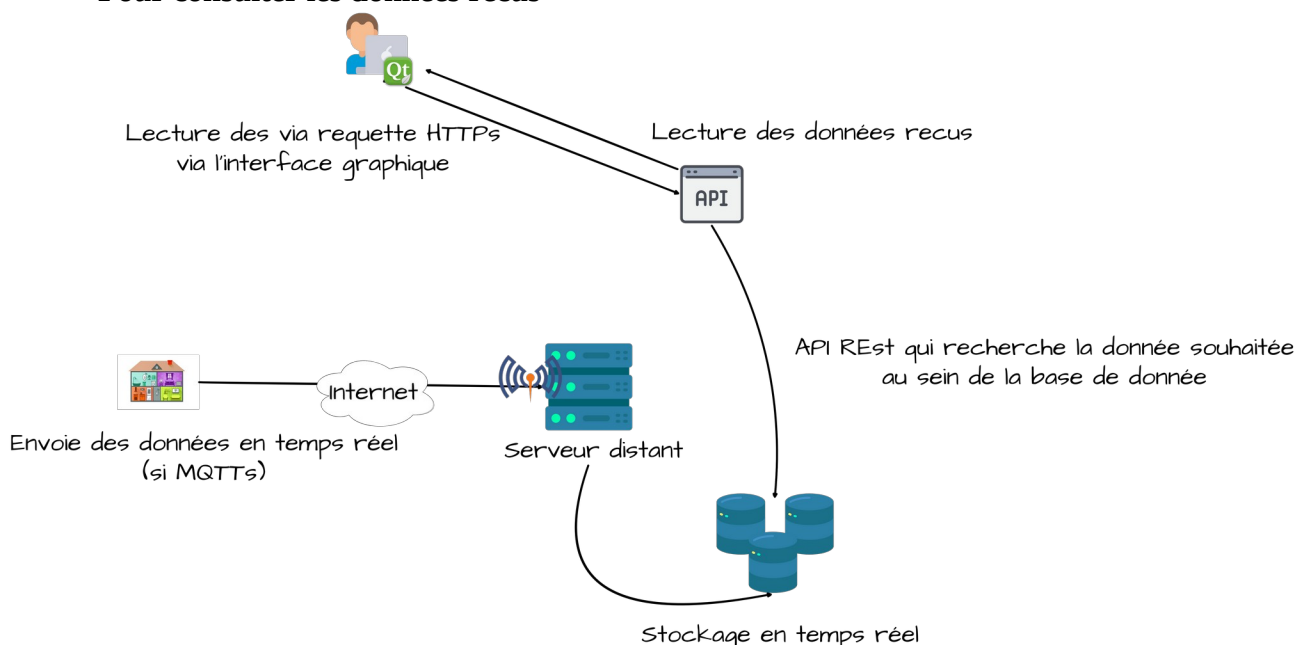
La conception d'une architecture réseau, l'implémentation d'automatisations et l'utilisation des dernières technologies IoT nous permettra d'acquérir une expérience concrète dans des domaines très recherchés et directement applicables à l'industrie. Enfin, ce projet constitue une excellente base pour une éventuelle extension future, que cela soit dans l'ajout de fonctionnalités ou dans l'intégration de nouveaux types d'appareils IoT.

Compétence qui seront acquises ou améliorés tout au long du projet:

- C++, Utilisation de **QT**.
- Framework permettant la création d'api Web (**Laravel** par exemple que j'ai utilisé à l'issue de mon stage, ou node js)
- Java + Android studio (si le temps permet de créer une application mobile)
- Intégration de Protocole réseau, **MQTT**, **HTTP** et **Websocket** pour la gestion des communication avec les appareils IOT.
- Sécurité réseau, **SSL/TLS** et hashage, pour la sécurisation des données.
- Base de données
- Outils de tests et simulateurs : **MQTT Simulators**, **IoTIFY**, **ThingSim**.
- Possibilité d'explorer **Docker** pour le déploiement et la conteneurisation, si on décide de créer nos simulateurs IOTs.

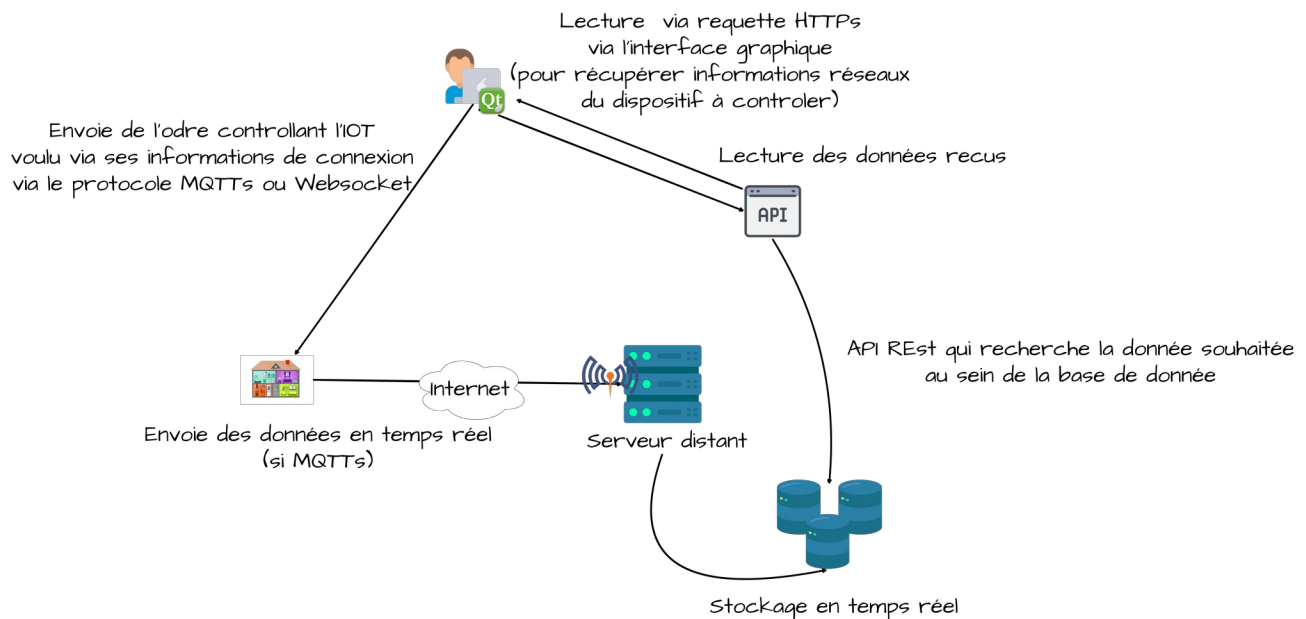
Voici une première version de l'architecture envisagé :

- Pour consulter les données recus



Dans cet exemple, la maison envoie en temps réel, à intervalles réguliers, l'état de ses différents dispositifs à un serveur distant en utilisant un protocole de communication adapté, tel que **MQTT** ou **WebSocket** pour assurer une transmission en temps réel. L'API REST est utilisée pour interagir avec la base de données : elle permet de lire les informations stockées et de répondre aux requêtes effectuées via des appels HTTP depuis l'application développée avec Qt. Ainsi, l'utilisateur peut bénéficier d'un **contrôle visuel en temps réel** de l'état de sa maison, avec la possibilité de surveiller les différents appareils et capteurs à distance.

- Contrôle distante de la maison:



Dans cet exemple, l'objectif serait d'utiliser d'abord le protocole HTTP pour se connecter à l'API et récupérer les informations de connexion et d'identification de l'appareil IoT cible (stockées dans la base de données). Ensuite, à partir du logiciel Qt, une requête MQTT ou WebSocket (le choix du protocole sera décidé pendant le projet) sera envoyée pour transmettre l'ordre souhaité à l'appareil IoT.