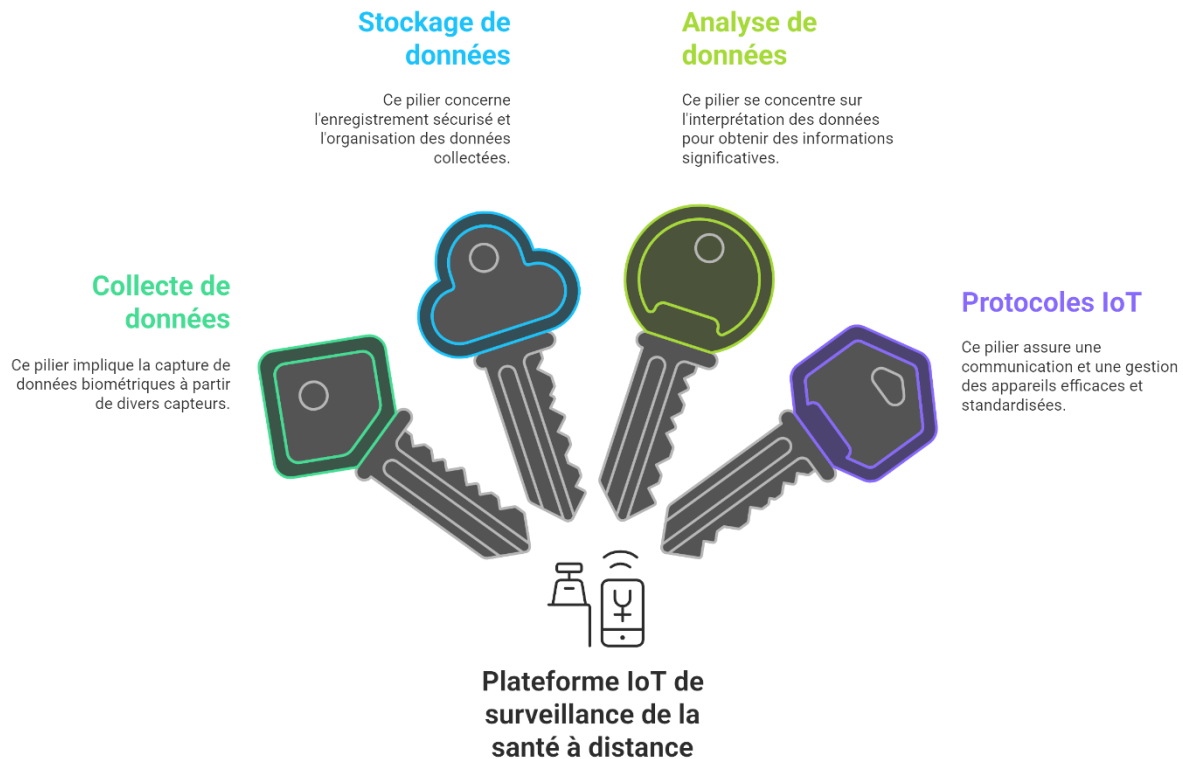


# Système de Surveillance de la Santé à Distance



## Objectif du Projet

L'objectif principal de ce projet est de développer une plateforme IoT de surveillance de la santé à distance. Cette plateforme permet de collecter, de stocker et d'analyser des données en temps réel provenant de capteurs biométriques (fréquence cardiaque, saturation en oxygène, etc.). Elle s'appuie sur des protocoles IoT standards pour une communication efficace et une gestion optimisée des appareils.

## Outils Utilisés

- **Backend**

- **\*\*Python (FastAPI)\*\*** : Développement de l'API REST pour gérer les données des capteurs et faciliter les interactions entre les différentes parties du système.
- **\*\*MongoDB Atlas\*\*** : Base de données NoSQL pour stocker les données des patients et des appareils de manière sécurisée et scalable.
- **\*\*Paho MQTT\*\*** : Implémentation des fonctionnalités en temps réel via le protocole MQTT pour la transmission rapide des données biométriques.

- Frontend

- **React.js** : Interface utilisateur interactive pour afficher les données des patients, les appareils connectés et les mises à jour en temps réel.
- **TypeScript** : Amélioration de la fiabilité et de la maintenabilité du code frontend.
- **Tailwind CSS** : Framework CSS pour un design moderne et responsive.

- Protocole IoT

- **MQTT** : Communication en temps réel entre les capteurs et la plateforme.
- **CoAP** : Optimisation des interactions avec les capteurs via des requêtes légères.
- **LwM2M** : Gestion des appareils connectés, incluant l'enregistrement, les mises à jour et la surveillance.

- Outils Supplémentaires

- **Eclipse Leshan** : Serveur LwM2M pour la gestion des appareils IoT.
- **Postman** : Test des API REST développées dans le backend.
- **GitHub** : Gestion du code source et collaboration.

## Difficultés Rencontrées

- **Intégration des Protocoles** : Adapter les protocoles IoT tels que LwM2M et CoAP pour fonctionner avec les solutions existantes.
- **Connexion en Temps Réel** : Résolution des problèmes de boucle d'événements lors de l'intégration de MQTT avec Python.
- **Déploiement de MongoDB Atlas** : Assurer une configuration sécurisée et optimiser les performances pour les données volumineuses.
- **Problèmes de compatibilité** : Des problèmes de compatibilité entre différentes versions des bibliothèques utilisées ont été rencontrés, nécessitant des ajustements et des mises à jour des dépendances.
- **Gestion des Tests** : Synchroniser les tests entre les composants backend et frontend tout en assurant une communication fluide avec les capteurs.

## Réalisations et Limitations

- Réalisations

- Implémentation d'un backend robuste avec FastAPI, MongoDB, et MQTT.
- Développement d'une interface utilisateur intuitive avec React.js, permettant l'affichage en temps réel des données des capteurs.
- Intégration réussie des protocoles MQTT et CoAP pour la communication IoT.
- Gestion des appareils IoT via LwM2M.

- Limitations

- Absence de tests sur un environnement réel avec plusieurs capteurs physiques.
- Fonctionnalités avancées de gestion des appareils via LwM2M (comme les mises à jour logicielles) encore à finaliser.

- Lien vers le Code Source

Le code source du projet est disponible sur GitHub : [https://github.com/golden-on/iot\\_systeme](https://github.com/golden-on/iot_systeme)