

ECOTRACK — Gestion de Plantes d'Intérieur

Projet Microservices — Présenté par Barry Ismaïla

Architecture distribuée Spring Boot + MySQL + Keycloak

■ Objectif du projet

Ecotrack est une application de suivi et gestion de plantes d'intérieur. Elle permet aux utilisateurs d'ajouter leurs plantes, suivre leur état de santé, recevoir des rappels d'entretien et planifier les arrosages automatiquement.

L'objectif est de créer une plateforme microservices simple, mais complète, respectant la grille d'évaluation : sécurité, communication, discovery, config, et déploiement Docker.

■ Exigences fonctionnelles et non-fonctionnelles

Exigences fonctionnelles :

- Ajouter / Modifier / Supprimer une plante.
- Programmer des rappels d'arrosage.
- Gérer les utilisateurs et leur authentification.
- Afficher la liste des plantes et leur état de santé.
- Notifications automatiques pour les rappels.

Exigences non-fonctionnelles :

- Architecture distribuée basée sur microservices.
- Sécurité avec Keycloak (JWT, rôles).
- Base de données MySQL.
- Tolérance aux pannes via Eureka et Config Server.
- Déploiement via Docker Compose.

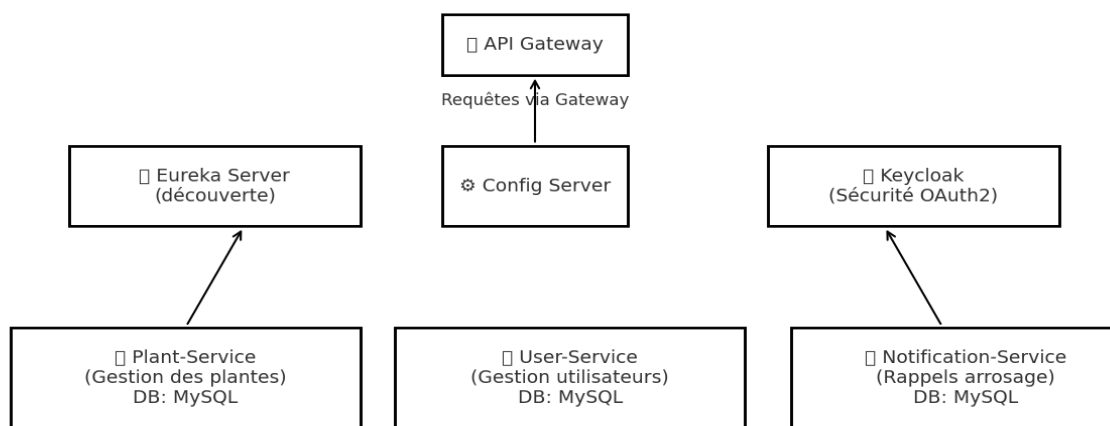
■■ Architecture microservices

L'architecture ECOTRACK repose sur quatre microservices principaux, complétés par les services d'infrastructure :

- ■ Plant-Service : gestion des plantes et de leurs états.
- ■ User-Service : gestion des utilisateurs et profils.
- ■ Notification-Service : planification et envoi des rappels.
- ■ API Gateway : point d'entrée sécurisé pour le front.

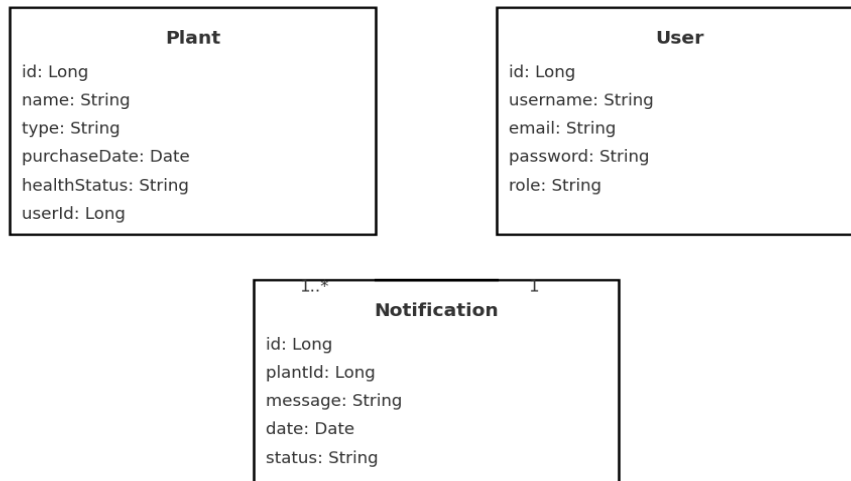
Infrastructure : Eureka (découverte), Config Server (config), Keycloak (sécurité).

Architecture microservices ECOTRACK



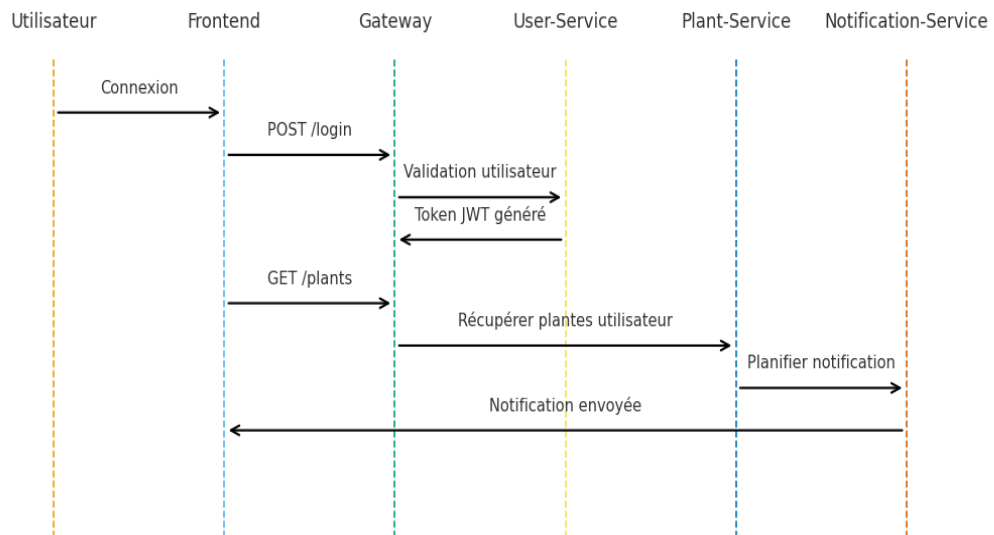
■ Diagramme de classes UML

Les principales entités de données du système : **Plant**, **User**, et **Notification**. Chaque utilisateur possède plusieurs plantes, et chaque plante peut avoir plusieurs notifications liées.



■ Diagramme de séquence UML

Ce diagramme illustre un scénario typique : un utilisateur se connecte, récupère ses plantes et reçoit des rappels programmés.



■ Lien avec la grille d'évaluation

- Microservices (Spring Boot + MySQL) : 1.5 pts.
- Eureka Discovery : 1 pt.
- Config Server : 1 pt.
- Gateway sécurisée (Keycloak/JWT) : 2 pts.
- Sécurité : 2 pts.
- Docker Compose complet : 2 pts.
- Frontend (Angular) : 1.5 pts.
- Communication inter-services : 2 pts.
- Valeur ajoutée (notifications automatiques) : 2 pts.

■ Étapes suivantes

1. Créer les projets Spring Boot pour chaque microservice.
2. Ajouter les Dockerfiles et le docker-compose.yml.
3. Configurer Keycloak et Eureka.
4. Développer un frontend Angular simple.
5. Tester la communication interservices et les notifications.

Présenté par Barry Ismaïla — Projet ECOTRACK ■