
MachinaEar

Systeme de Maintenance Prédictive Intelligent

Élaboré par :

Aziz BENSALAH
Salem Aziz FRADI
Zied KALLEL
Chiheb Eddine ZIDI

Encadré par :

Dr. Mohamed Béchaa
KAÂNICHE

1 Introduction

MachinaEar est un projet de maintenance prédictive embarquée visant à surveiller la santé des machines industrielles à faible coût. Le système repose sur un **Raspberry Pi 4** équipé de capteurs acoustiques et vibratoires, un modèle d'intelligence artificielle embarqué (**TensorFlow Lite**), et une architecture logicielle cloud basée sur **Jakarta EE** et **MongoDB Atlas**.

L'objectif est de détecter précocement les anomalies mécaniques, de prévenir les défaillances, et d'offrir une interface web intuitive pour la supervision et la maintenance proactive.

—

2 Architecture Générale du Système

L'architecture de MachinaEar repose sur trois couches interdépendantes :

- **Couche Edge (IoT)** : Raspberry Pi 4 exécutant un modèle TensorFlow Lite pour l'analyse locale des signaux.
- **Couche Cloud** : Serveur WildFly hébergeant les services REST Jakarta EE et la base de données MongoDB Atlas.
- **Couche Application** : Interface web Jakarta EE permettant la visualisation en temps réel, la configuration et la gestion des alertes.

—

3 Diagramme de Cas d'Utilisation

Le diagramme suivant illustre les interactions entre les acteurs et les principaux cas d'utilisation du système.

Description des acteurs

- **Utilisateur** : surveille la santé des machines, consulte les historiques et reçoit les alertes.
- **Administrateur** : gère les utilisateurs et les accès.
- **Raspberry Pi** : collecte et envoie les données capteurs au cloud.
- **Serveur Cloud** : stocke les données, entraîne et met à jour le modèle IA.

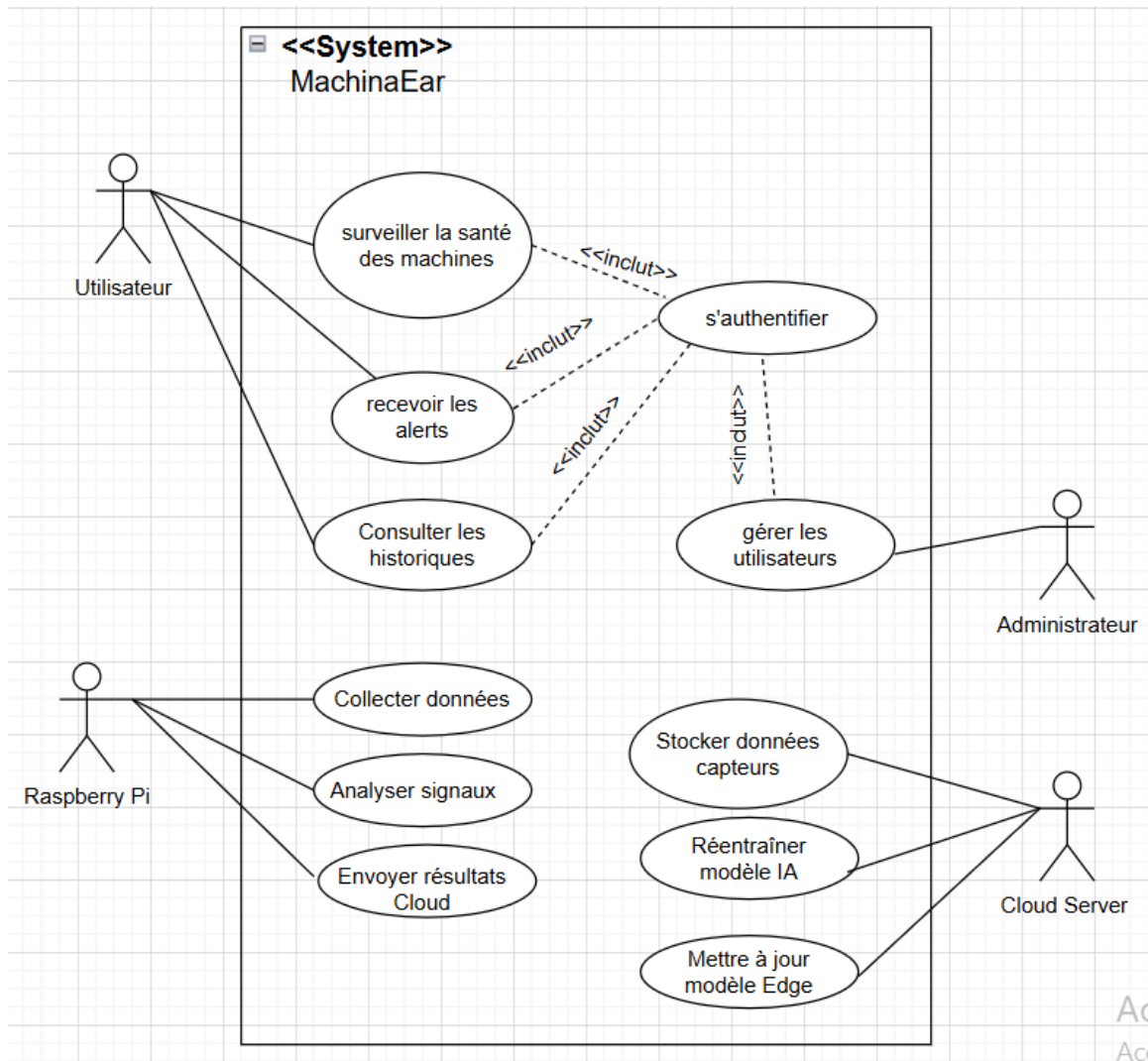


FIGURE 1 – Diagramme de cas d'utilisation du système MachinaEar.

4 Diagramme de Classes

Ce diagramme présente la structure orientée objet du système, les classes principales et leurs relations.

Analyse des relations principales

- **Administrateur** hérite de la classe **Utilisateur**.
- Un **Utilisateur** consulte un **TableauDeBord** (1→1).
- Le **TableauDeBord** affiche l'état de plusieurs **Machines**.
- Une **Machine** contient un ou plusieurs **Capteurs**.
- Les **Capteurs** envoient leurs données au **DétecteurAnomalie** via un **MQTT-Client**.
- Le **DétecteurAnomalie** génère des **Alertes** et met à jour son modèle via le

ServeurCloud.

- Le **ServeurCloud** stocke les données dans le **DatabaseManager**.

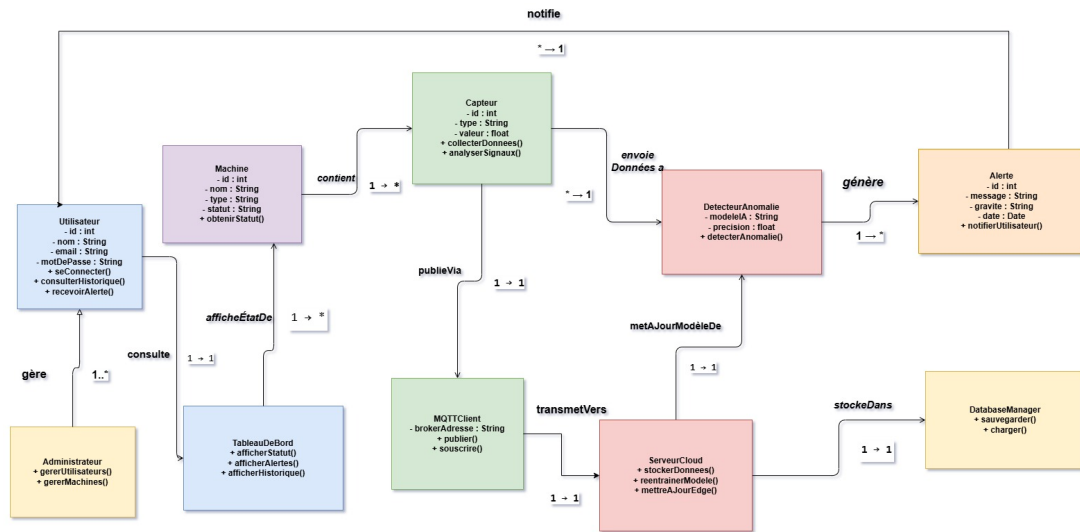


FIGURE 2 – Diagramme de classes du système MachinaEar.

5 Diagramme de Déploiement

Le diagramme suivant illustre la distribution physique du système entre les couches Edge, Cloud et IA.

Description du déploiement

- **Raspberry Pi (Edge)** : exécute le modèle TensorFlow Lite et publie les mesures via MQTT (Eclipse Mosquitto).
- **Serveur Cloud (WildFly)** : héberge les API REST Jakarta EE et interagit avec MongoDB Atlas.
- **Pipeline IA** : GitHub Actions et MLflow assurent le réentraînement et la mise à jour automatique des modèles.
- **MongoDB Atlas** : stocke les données capteurs, historiques et alertes.

6 Sécurité et Communication

- Transmission sécurisée via **HTTPS** et **MQTTs**.
- Authentification par jeton **JWT**.
- Contrôle d'accès basé sur les rôles (utilisateur / administrateur).
- Sauvegarde automatique des données et versioning des modèles IA.

7 Améliorations Futures

- Intégration d'une application mobile Android connectée à l'API REST.
- Ajout de nouveaux capteurs : température, courant, pression.
- Support du protocole industriel OPC-UA.
- Visualisation enrichie via PrimeFaces et JSF.

8 Conclusion

4