

MicroLearn

Table des matières

1. Diagramme BPMN – Processus métier de MicroLearn	3
1.1. Processus métiers principaux.....	3
1.2. Architecture microservices.....	3
2. Conception détaillée microservices	5
2.1. Micro1 : DataPreparer	5
Diagramme de classes.....	5
Diagramme de cas d'utilisations	5
2.2. Micro2 : ModelSelector	6
Diagramme de classes.....	6
Diagramme de cas d'utilisations	6
2.3. Micro3 : HyperOpt	7
Diagramme de classes.....	7
Diagramme de cas d'utilisations	7
2.4. Micro4 : Trainer	8
Diagramme de classes.....	8
Diagramme de cas d'utilisations	8
2.5. Micro5 : Evaluator.....	9
Diagramme de classes.....	9
Diagramme de cas d'utilisations	9
2.6. Micro6 : Deployer.....	10
Diagramme de classes.....	10
Diagramme de cas d'utilisations	10
2.7. Micro7 : Orchestrator	11
Diagramme de classes.....	11
Diagramme de cas d'utilisations	11
3. Maquettes UI/UX.....	12
3.1. Dashboard: Vue globale des pipelines avec timeline.	12

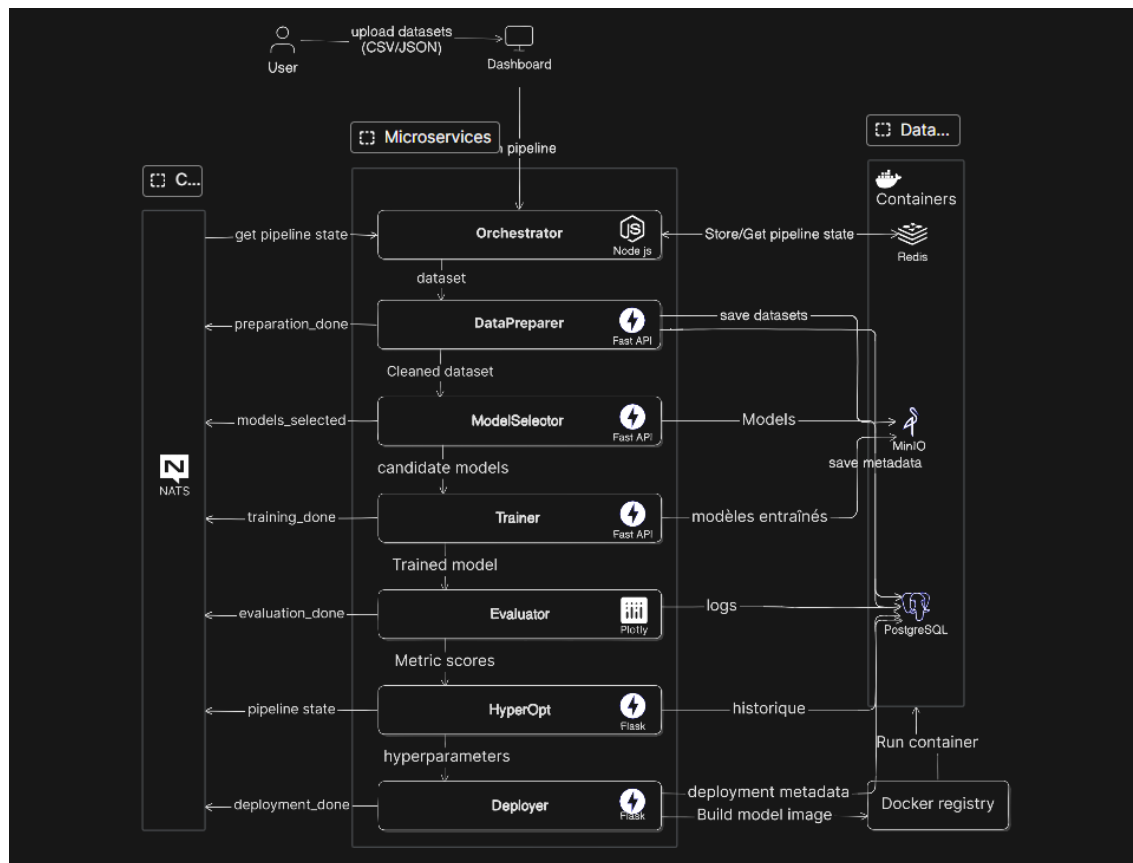
1. Diagramme BPMN – Processus métier de MicroLearn

BPMN (Business Process Model and Notation) permet de représenter visuellement les étapes métier de ton système. Pour MicroLearn, on peut définir le flux suivant :

1.1. Processus métiers principaux

Microservice	Role
Upload Dataset	L'utilisateur téléverse un CSV/JSON.
Data Preparer	Nettoyage, normalisation, encodage.
Model Selector	Sélection automatique des modèles candidats.
HyperOpt	Optimisation des hyperparamètres.
Trainer	Entraînement parallèle des modèles optimisés.
Evaluator	Évaluation complète et calcul des métriques.
Deployer	Déploiement automatique des meilleurs modèles.
Dashboard	Visualisation et suivi des pipelines

1.2. Architecture microservices



Résumé microservices :

Microservice	Rôle	Tech	DB/Stockage	Communication
DataPreparer	Nettoyage & préparation des données	Python, Pandas, FastAPI	PostgreSQL, MinIO	Asynchrone
ModelSelector	Sélection modèles ML	scikit-learn, PyCaret, FastAPI	PostgreSQL	Asynchrone
HyperOpt	Optimisation hyperparamètres	Optuna, Redis, FastAPI	Redis	Asynchrone
Trainer	Entraînement parallèle GPU	PyTorch Lightning, Ray, MLflow	PostgreSQL	Asynchrone
Evaluator	Évaluation modèles	scikit-learn, Plotly	PostgreSQL	Asynchrone
Deployer	Déploiement REST/BATCH	TorchServe, Flask, Docker	—	REST / Synchrone
Orchestrator	Orchestration pipeline	Node.js, NATS, Redis	Redis	Synchrone/Asynchrone
Dashboard	Visualisation & suivi	React, D3.js, Chart.js	PostgreSQL, MinIO	REST / WebSocket

2. Conception détaillée microservices

2.1. Micro1 : DataPreparer

Diagramme de classes

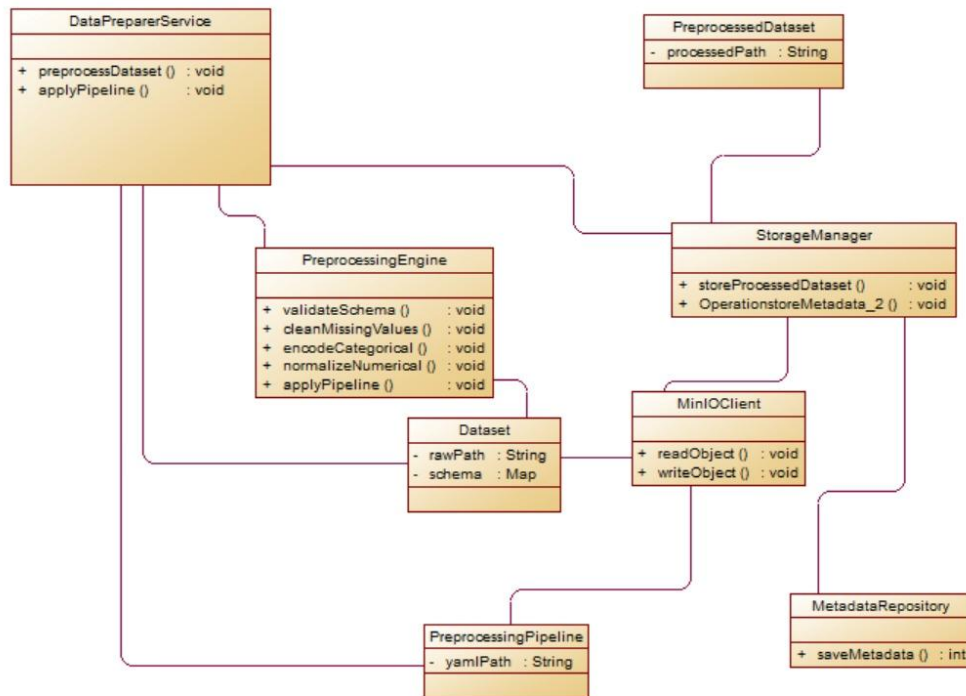
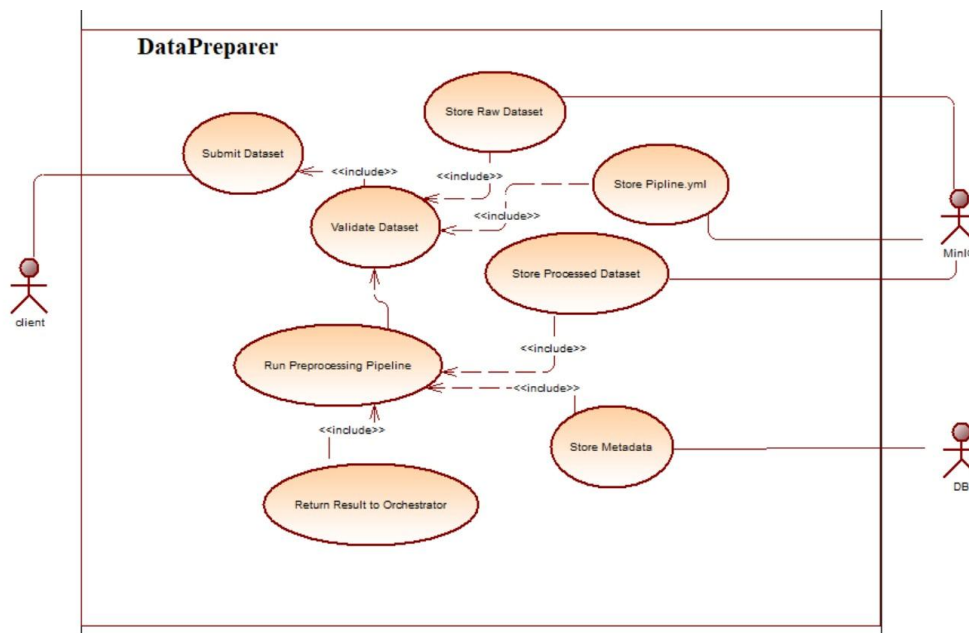


Diagramme de cas d'utilisations



2.2. Micro2 : ModelSelector

Diagramme de classes

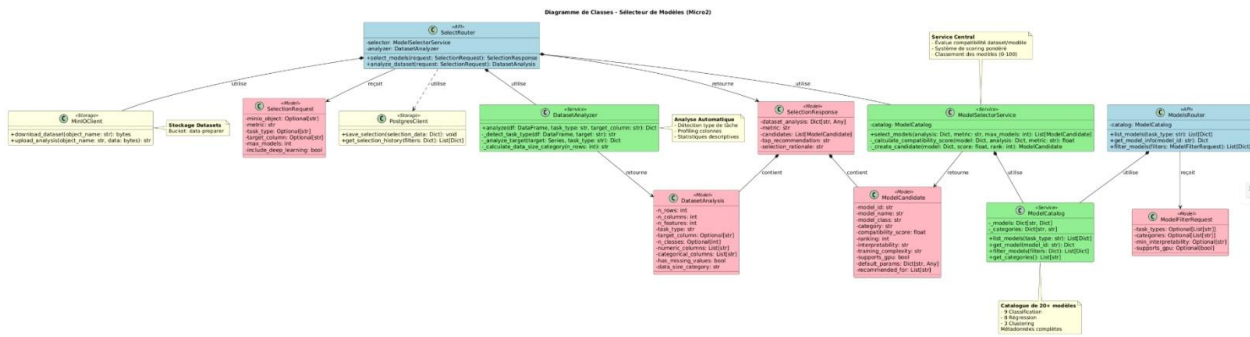
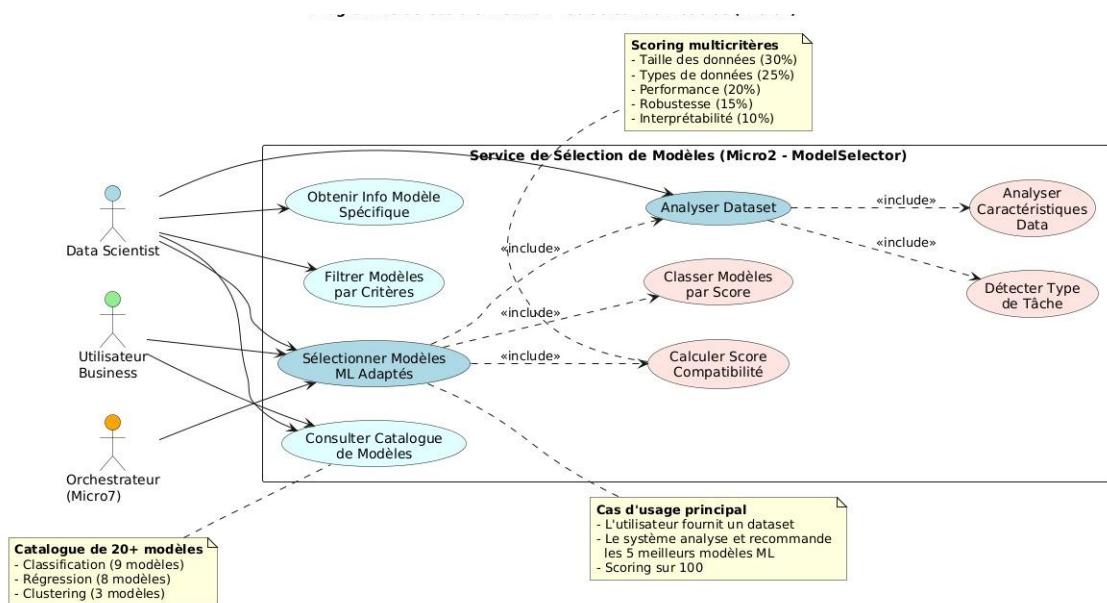


Diagramme de cas d'utilisations



2.3. Micro3 : HyperOpt

Diagramme de classes

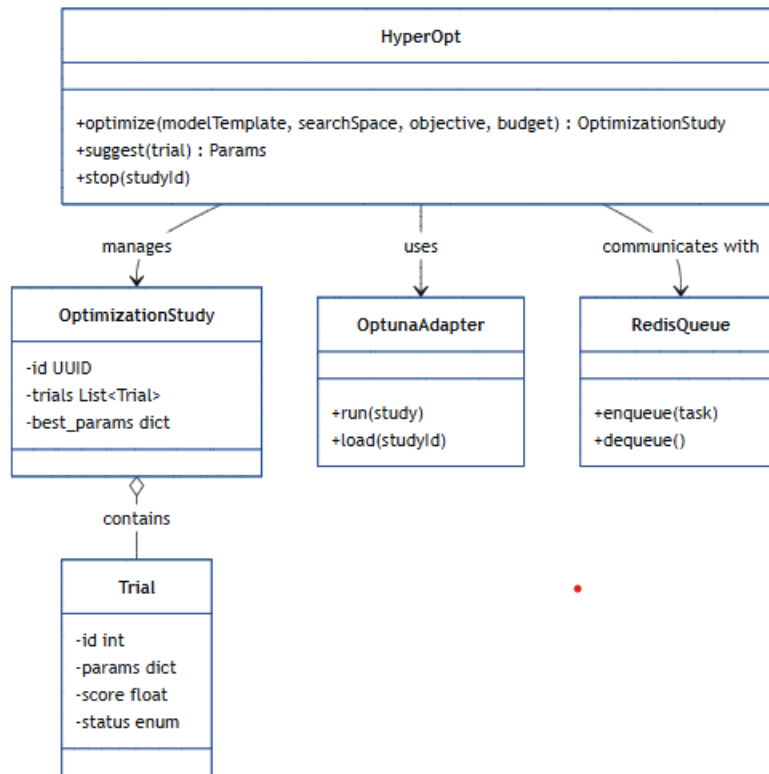
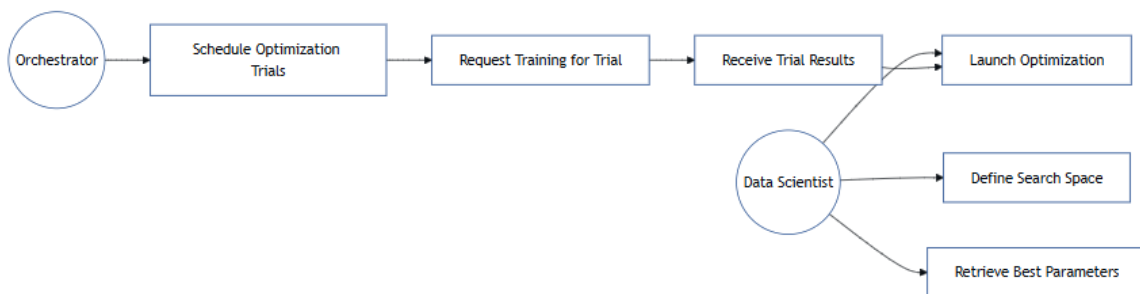


Diagramme de cas d'utilisations



2.4. Micro4 : Trainer

Diagramme de classes

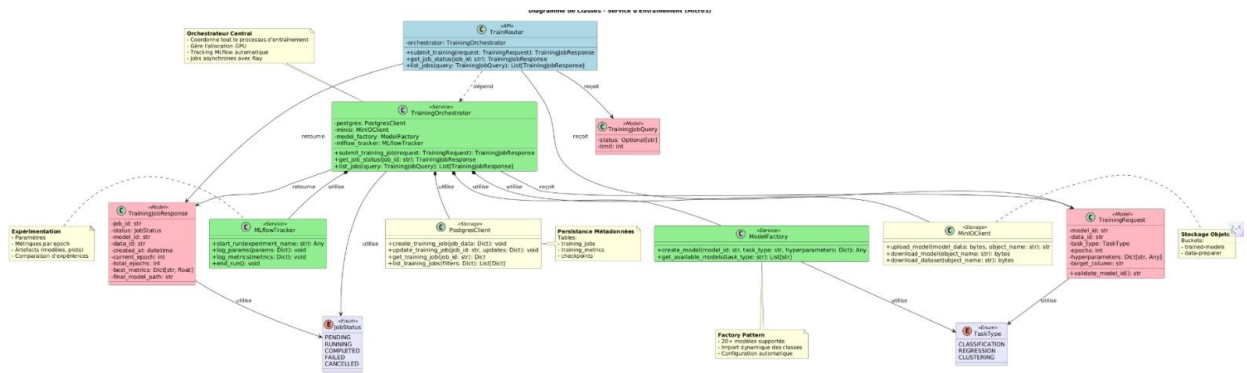
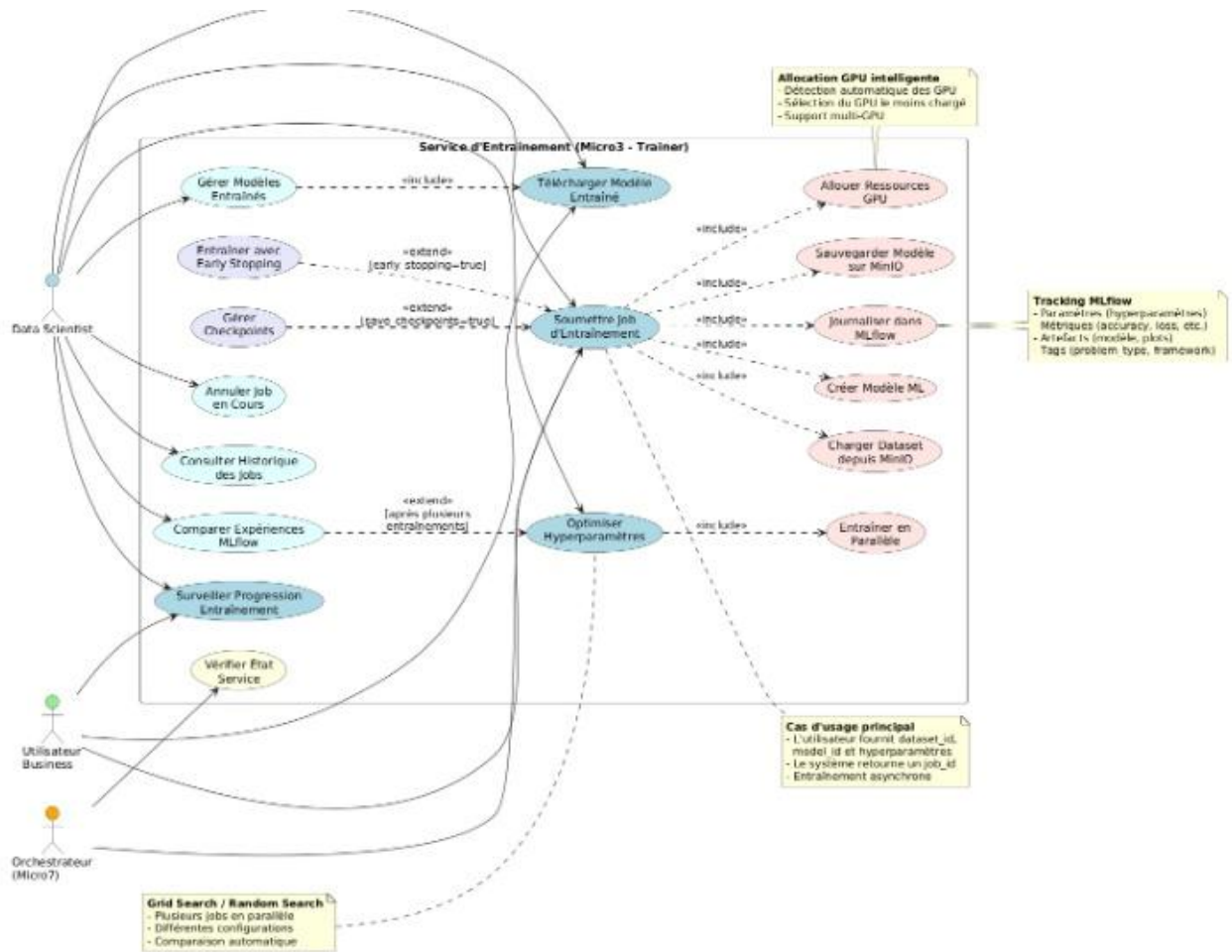


Diagramme de cas d'utilisations



2.5. Micro5 : Evaluator

Diagramme de classes

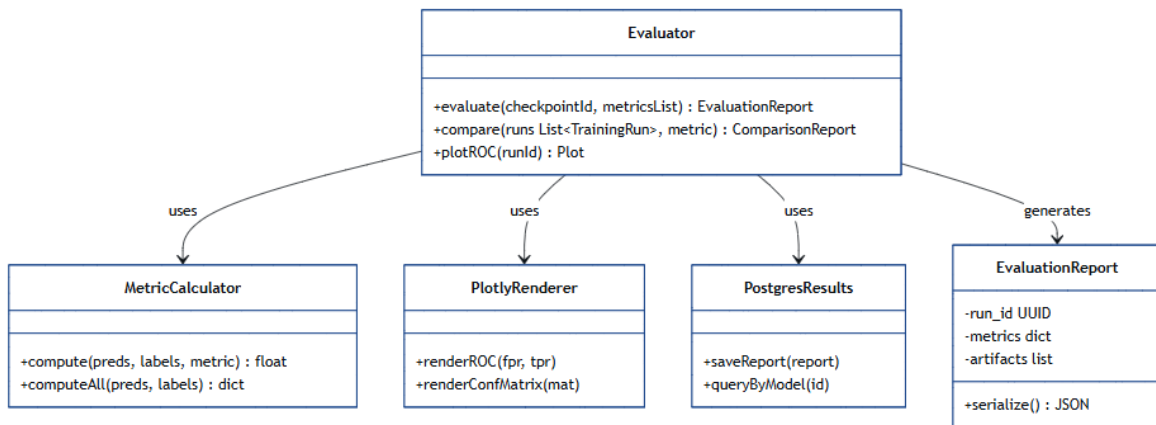
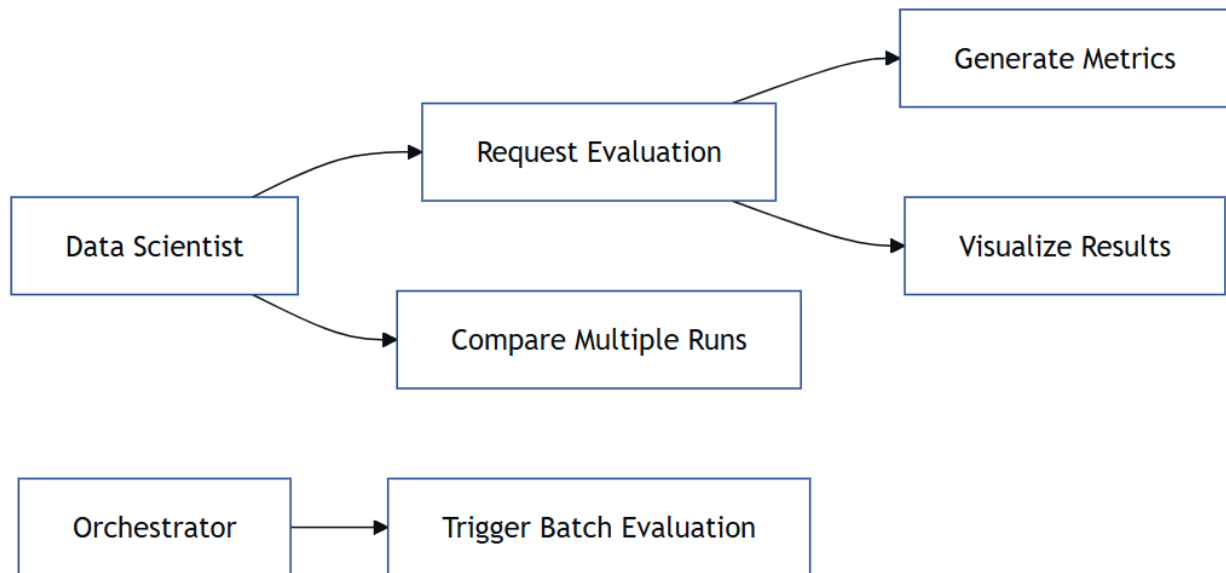


Diagramme de cas d'utilisations



2.6. Micro6 : Deployer

Diagramme de classes

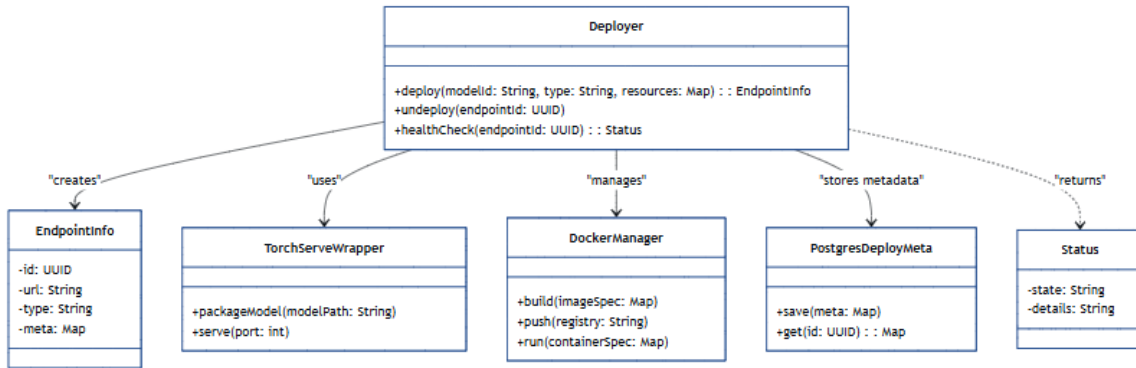
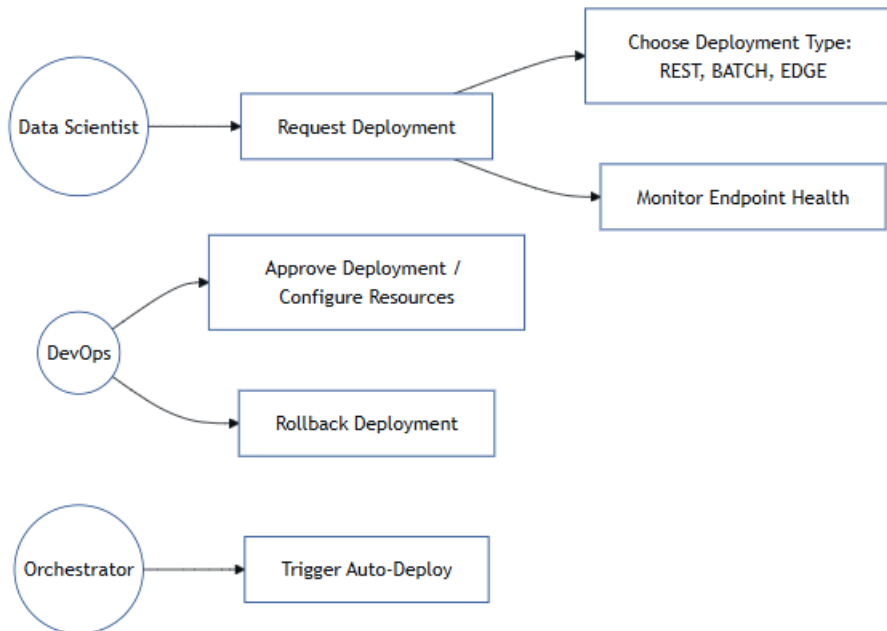


Diagramme de cas d'utilisations



2.7. Micro7 : Orchestrator

Diagramme de classes

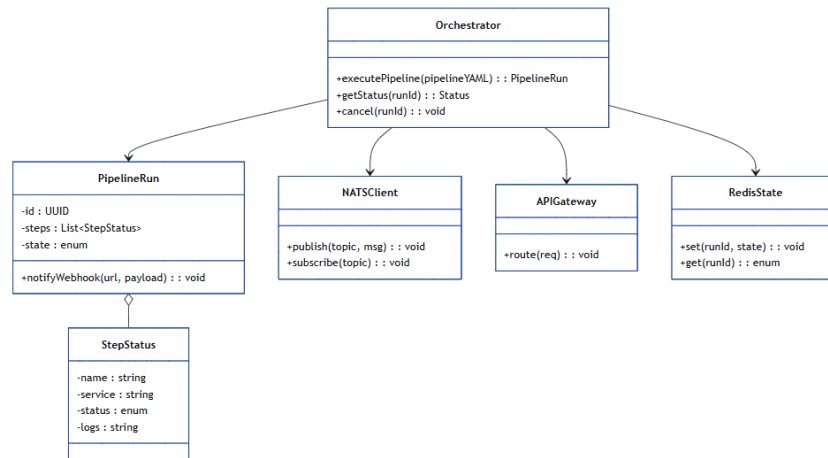
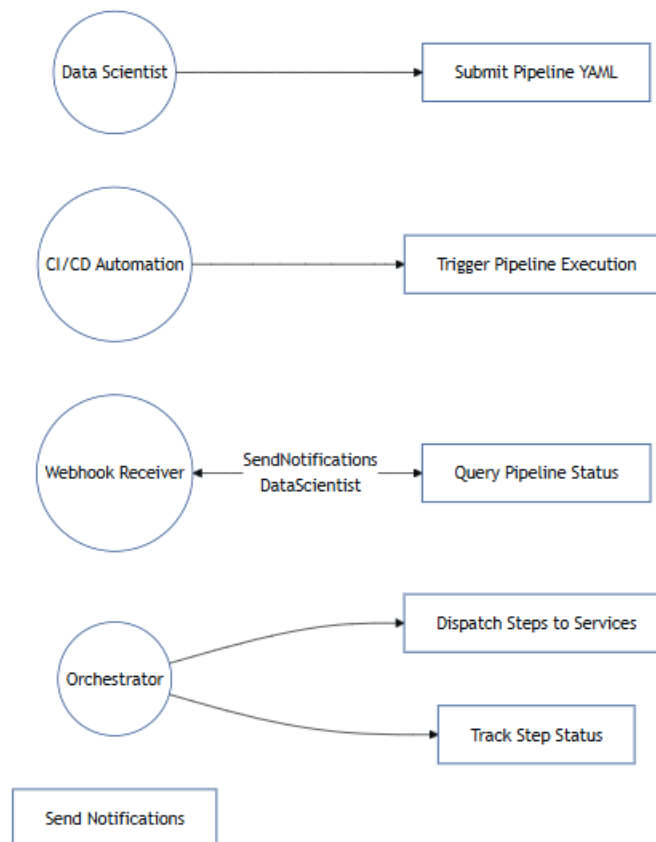


Diagramme de cas d'utilisations



3. Maquettes UI/UX

Lien Figma : <https://www.figma.com/design/MJXTsysh2zdfqlDVpU3bu9/MicroLearn?node-id=0-1&t=T9loFXs4YzxR1ap7-1>

3.1. Dashboard: Vue globale des pipelines avec timeline.