

Point Intermédiaire

Les technologies utilisées

- **Backend**

- Databases

- MongoDB (Colis, Suivi de trajet, metrics)
 - LightSQL (User Management, Rights Managements)

- Serveur

- API
 - FastAPI
 - Client MongoDB
 - PyMongo
 - Motor
 - AsyncIO
 - Motor-stubs
 - Client SQL
 - asyncpg (TODO)

- Deploiement

- Docker

Note:

Notamment, avec l'utilisation de Docker Compose.

- **Frontend (TODO)**

- Serveur Web (TODO)

- NextJS

- Client Python (TODO)

- Websocket (Python Package)

- Langage

- ReactJS
- TypeScript

• Environnement de test

L'environnement partagé entre le backend et le frontend, permettant d'avoir un "miroir", au niveau fonctionnel de ce que chaque composant peut attendre de l'autre.

- Backend Replicata
 - Docker
 - MongoDB For Community
 - LightSQL
 - Async Python Server for Testing
- FrontEnd Replicata
 - NextJS Server
 - Docker
 - Python Client for Testing

Partage des tâches

Le projet est partagé en deux parties distinctes où Alexis Opolka s'occupe principalement du Backend et Mathys Domergue s'occupe principalement du Frontend, cependant, il faut noter que l'on ne se limite pas à cette délimitation et Alexis et Mathys travaillent sur quelques aspects de chaque partie.

Temps travaillé

Alexis a fait une première version du backend fonctionnel avec que du MongoDB, synchrone et une API REST en 6-7h.

Il refait toutefois cette partie en y ajoutant du SQL, de l'asynchrone et le support pour des connexions WebSockets.

Mathys a fait un PMV (Produit Minimal Viable) d'un client Python ainsi que le début d'une interface Web en 15-20h.

Ce qui nous reste à faire

Il reste à passer les PMV en rendu fonctionnel, finir l'environnement de test et déployer le tout en docker compose dans les prochains jours.

