

Architecture Logicielle: Construction

Equipe B: Guillaume Piccina, William d'Andrea, Nicolas Fernandez, Yann Brault

Sujet V6 : Pay as you pollute

Ce document traite de l'architecture et de son évolution chaque semaine.

Semaine 41: 11_10

Comme prévu dans le programme de la semaine dernière, l'architecture a été repensée cette semaine.

La majorité du découpage reste le même mais les responsabilités, les connexions aux services externes et le flot de données ont été modifiés.

- Les services car tracker, tracking shutdown, billing ainsi que la DB principale et le service mettant à disposition les informations relatives aux zones de pollution ont été conservés.
- En revanche, le service payment a lui été supprimé.
- Les relations avec les services extérieurs ont été modifiées.

En effet c'est maintenant le service Car tracker qui interroge le service de pollution afin d'obtenir lui même la zone dans laquelle se trouve l'utilisateur en se basant sur la position gps qu'il envoie.

Un nouveau service externe qui entre dans l'architecture est le service de surveillance des caméras de la ville qui est connecté directement au service Car tracker et qui interviendra dans le cadre d'une vérification contre des comportements frauduleux.

Les responsabilités de chaque service ont également été revues car certains comme Billing en avaient trop et parfois non justifiés, ainsi voici un détail de fonctionnement des services:

- Car tracker: Reçoit les informations de tracking de l'utilisateur, il s'occupe de faire une correspondance entre la position gps et la zone de pollution par l'intermédiaire du service des zones de pollution, il peut faire une vérification vidéo si c'est jugé utile et puis transmet les informations à la base de données.
- Tracking shutdown: Ce service n'a pas été modifié, il est là pour faire un relais entre l'user et billing pour prévenir d'une fin de trajet.

- Billing: Ce service avait trop de responsabilités, donc certaines lui ont été enlevées. Ainsi c'est maintenant le car tracker et le service de pollution qui s'occupe de garder, mettre à jour et faire la correspondance position/ zone car il a été jugé trop sensible de laisser l'utilisateur faire ce parallèle. Autre changement, le calcul d'estimation de la facture (pour le coût en temps réel) a été déporté sur le terminal de l'utilisateur. Ce changement est jugé moins critique car c'est le service de billing et le calcul qu'il fera qui fera foi et sera pris comme source unique de vérité quant au prix d'un trajet. Donc le service de billing reçoit des notifications de fin de trajet de la part du service de tracking shutdown, récupère dans la base de données les informations relatives au user qui a fini son trajet, calcul le coût final et fait une requête de paiement aux services extérieurs des banques.

On distingue donc 3 services externes qui sont le service pollution de la ville, le service de caméra de surveillance et les services des banques.

Scénario du flot de données:

- L'utilisateur envoie son id, sa position gps et un timestamp au service de tracking, il reçoit sa zone de circulation en réponse à sa requête ce qui lui permet de calculer une estimation du coût de son trajet
- Le service de tracking envoie la position gps au service de pollution qui lui renvoie une zone de pollution, puis le tracker envoie les 4 informations au stockage.
- Quand l'utilisateur finit son trajet, il envoie les mêmes informations que d'habitude sauf que cette fois-ci il les envoie au service tracking shutdown
- Le service tracking shutdown notifie le service de billing en lui transmettant l'id de l'utilisateur qui a fini sa course
- Le service de billing va aller pull dans la base de données toutes les informations de tracking relatif à l'id qu'il a reçu afin de calculer la facture puis il va envoyer la facture au service de banque
- L'utilisateur recevra sa facture directement de sa banque

Enfin voilà un schéma de l'architecture:

