



# Spécifications techniques

**Auteur :** Sita Béréte

**Version :** 1.0

**Dernière mise à jour :** 20 mai 2021

# Sommaire

<b>1. Choix technologiques</b>	<b>3</b>
<b>2. Nom de domaine et Hébergement</b>	<b>3</b>
<b>3. Domaine fonctionnel</b>	<b>3</b>
<b>4. Diagramme de composant</b>	<b>5</b>
<b>5. Diagramme de déploiement</b>	<b>6</b>

# 1. Choix technologiques

Pour la réalisation du site web et de l'espace d'administration, Nous utiliserons le langage de programmation **PHP** et le **SGBD** (Système de Gestion de Base de Donnée) **Postgresql**, parce que nos développeurs ont de l'expérience sur ces dernières, mais aussi parce qu'ils sont open-source, ils existent depuis plusieurs années et il sont très utilisés (ils ont donc de grosses communautés).

Quant au développement de l'application des livreurs, nous utiliserons le framework Flutter, ce qui nous permettra de rendre l'application disponible sur Android et les autres sans augmenter le temps de développement ni le coût du projet.

Et pour finir avec les choix technologiques, nous utiliserons le langage de programmation Java pour développer l'application des pizzaiolos, puisqu'on a pas besoin que celle-ci soit multiplateforme.

# 2. Nom de domaine et Hébergement

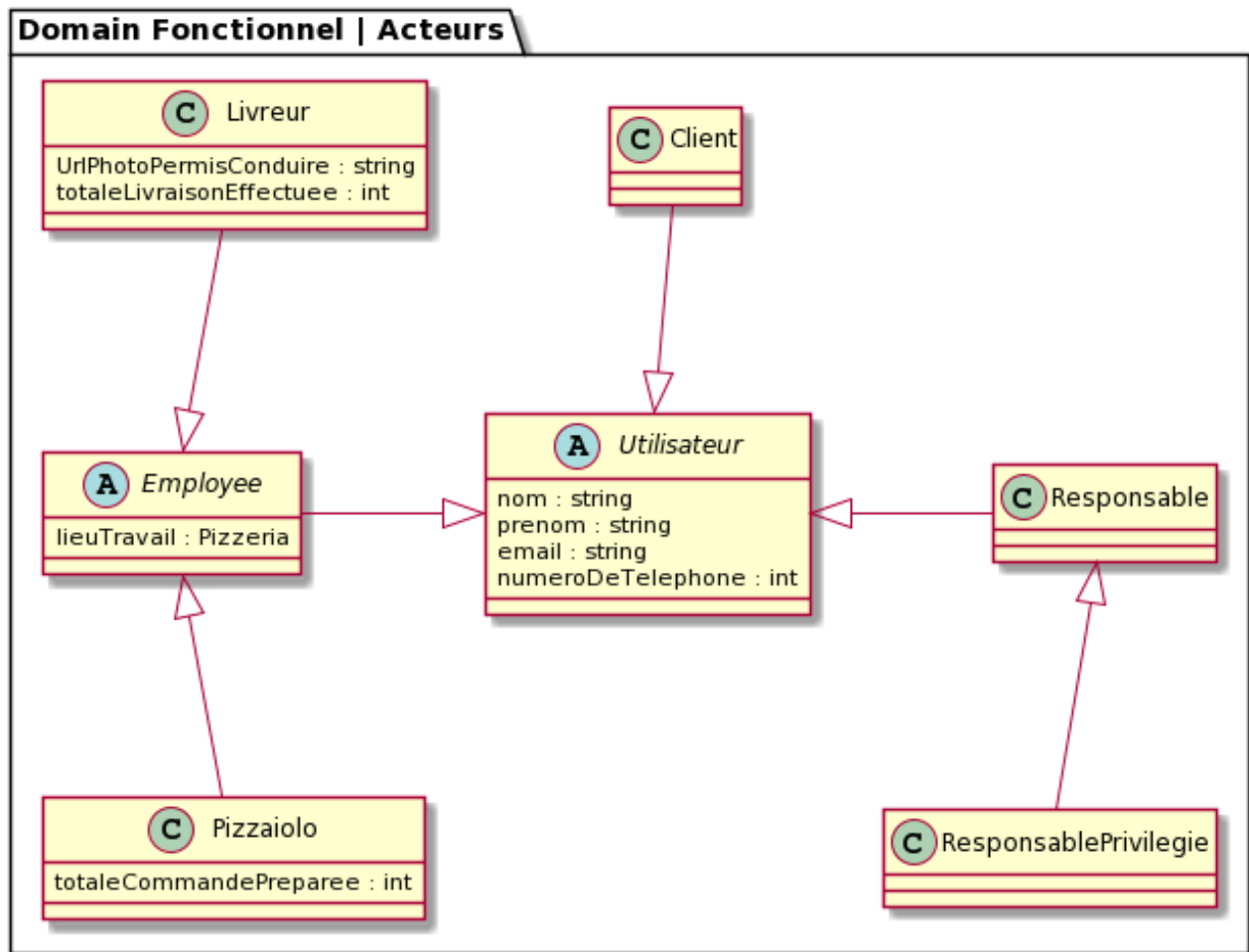
Le nom de domaine [oc-pizza.com](https://oc-pizza.com) sera réservé pour le site et l'espace d'administration.

Quant à l'hébergement nous utiliserons des services cloud, ce qui nous permettra de ne nous occuper que d'une petite partie du côté DevOps et laisser tout le reste au fournisseur de service cloud. En se basant sur l'expérience qu'ont nos développeurs, nous utiliserons deux services fournis par [GCP](#) (Google Cloud Platform) : [App Engine](#) pour l'hébergement de l'application backend et [Cloud SQL](#) pour la base de données.

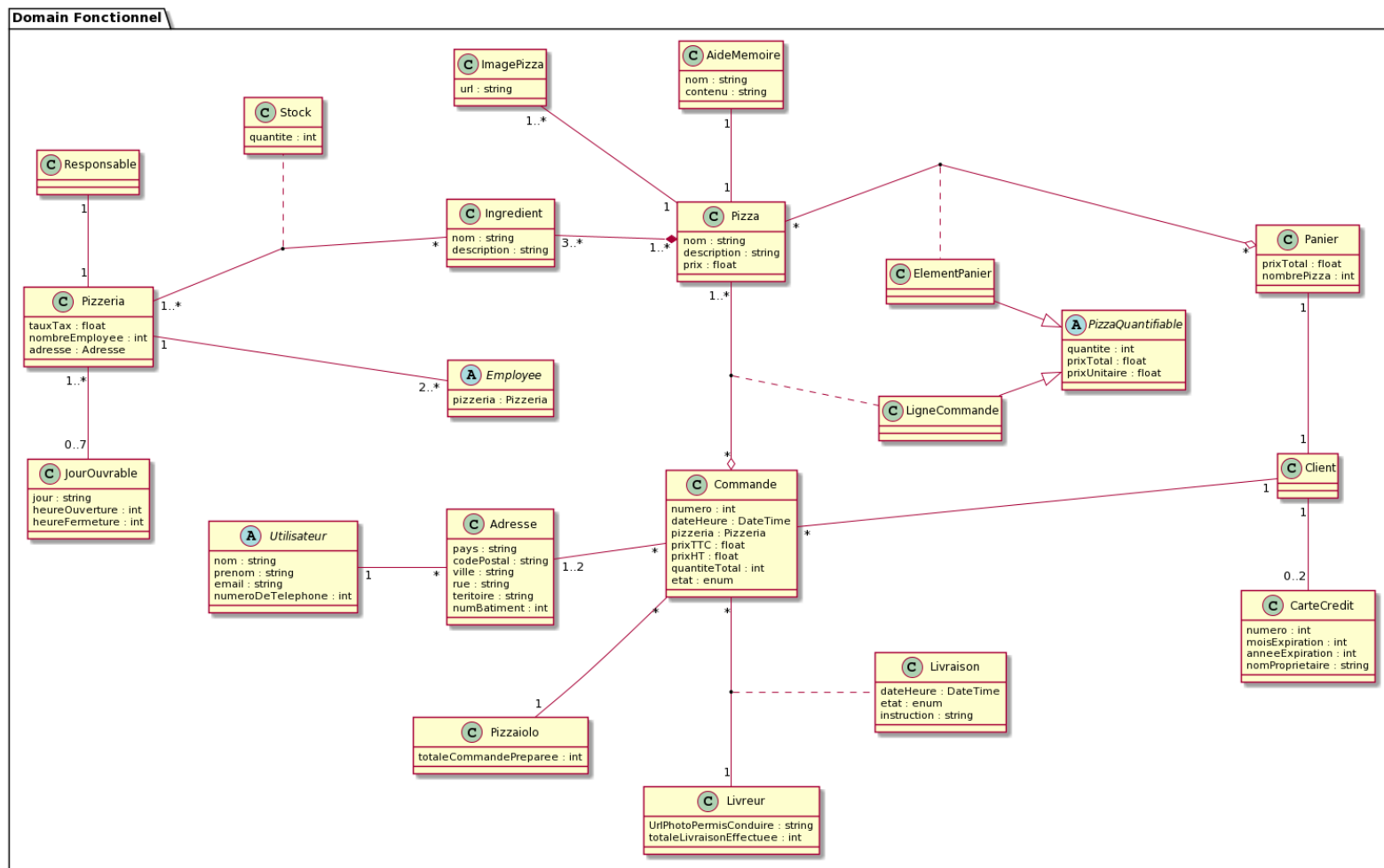
# 3. Domaine fonctionnel

Pour limiter le nombre d'associations dans le domaine fonctionnel et ainsi améliorer la lisibilité de ce dernier, il est séparé en deux diagrammes :

1. Le diagramme de classe montrant les différentes relations qui existent entre les classes représentant seulement les acteurs,

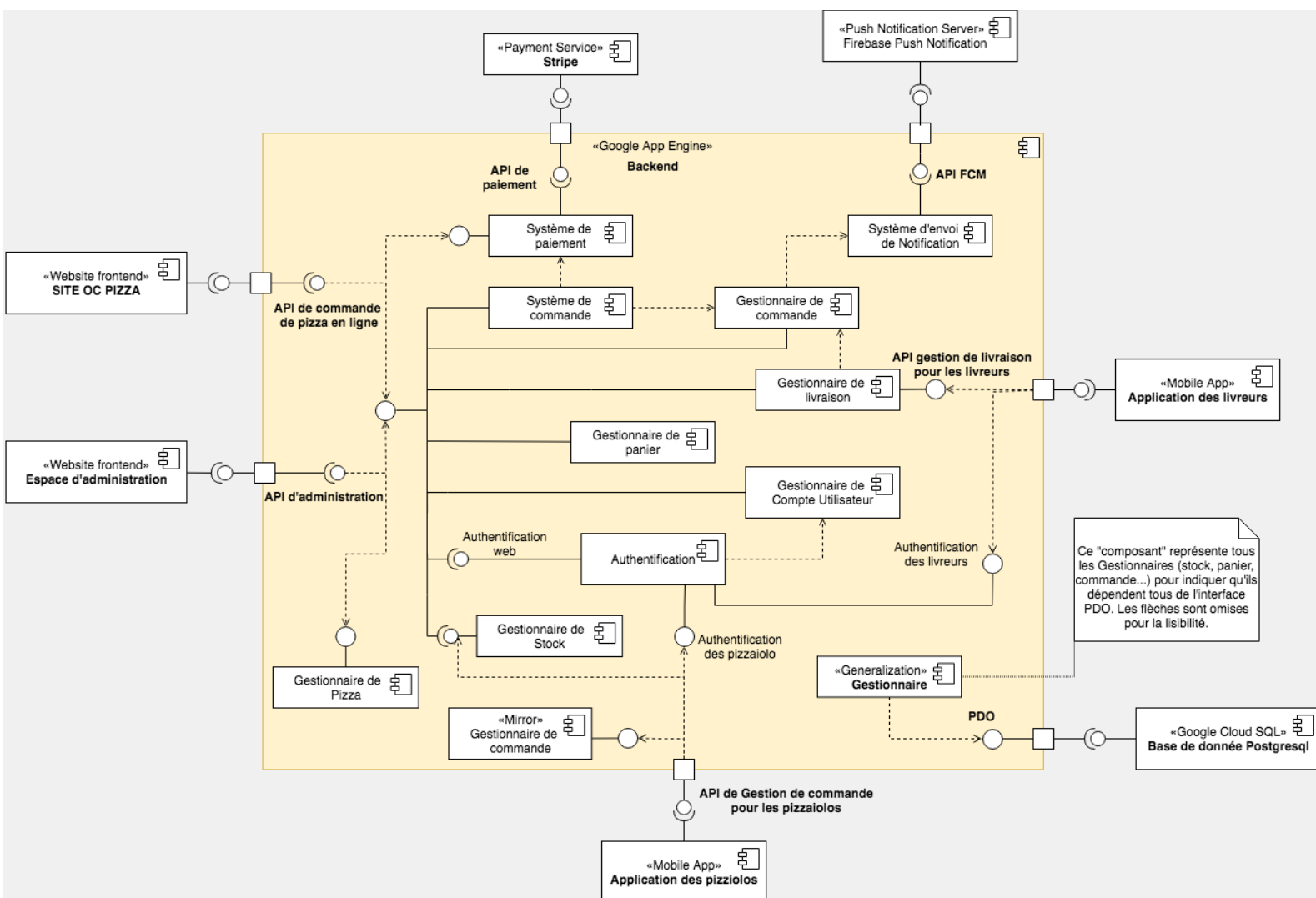


2. On a ensuite le diagramme de classe représentant tout le domaine fonctionnel du système, mais sans représentation des relations entre les Acteurs.



## 4. Diagramme de composant

Le diagramme ci-dessous représente toutes les relations qui existent entre les futurs composants du système, les services tiers dont le système aura besoin pour fonctionner, ainsi que les interfaces liant notre système à ces derniers.



## 5. Diagramme de déploiement

Le diagramme ci-dessous représente les différents composants matériels sur lesquels seront déployés nos futurs sous-systèmes, on y voit aussi les moyens de communications qui seront utilisés pour les faire communiquer.

