OC Pizza

ERP OC Pizza

Dossier de conception technique

Version 1.0

Auteur
Etienne BARBIER
Développeur

TABLE DES MATIERES

EB Consulting 1 rue du Pont, 75000 Paris – 01.02.03.04.05 – contact@eb-consulting.com
www.eb-consulting.com S.A.R.L. au capital de 1 000,00 € enregistrée au RCS de Paris – SIREN 999 999 999 – Code APE : 6202A

1 - VERSIONS

Auteur	Date	Description	Version
Etienne BARBIER	04/05/2020	Création du document	1.0

EB Consulting 1 rue du Pont, 75000 Paris – 01.02.03.04.05 – contact@eb-consulting.com
www.eb-consulting.com S.A.R.L. au capital de 1 000,00 € enregistrée au RCS de Paris – SIREN 999 999 999 – Code APE : 6202A

2 - INTRODUCTION

2.1 - Objet du document

Le présent document constitue le dossier de conception technique de l'application « ERP OC Pizza ».

L'objectif de ce document est de décrire l'architecture technique, l'architecture de déploiement et l'architecture logicielle de l'application « ERP OC Pizza ».

Les éléments du présent dossier découlent d'échanges de courriels avec OC Pizza et du dossier de conception fonctionnelle de l'application « ERP OC Pizza ».

2.2 - Références

Pour de plus amples informations, se référer également aux éléments suivants :

1. **PDOCPizza_01_dossier_conception_fonctionnelle** : dossier de conception fonctionnelle de l'application
2. **PDOCPizza_03_dossier_exploitation** : dossier d'exploitation de l'application

EB Consulting 1 rue du Pont, 75000 Paris – 01.02.03.04.05 – contact@eb-consulting.com
www.eb-consulting.com S.A.R.L. au capital de 1 000,00 € enregistrée au RCS de Paris – SIREN 999 999 999 – Code APE : 6202A

3 - LE DOMAINE FONCTIONNEL

3.1 - Référentiel : diagramme de classes

Voici le diagramme de classes de l'application :

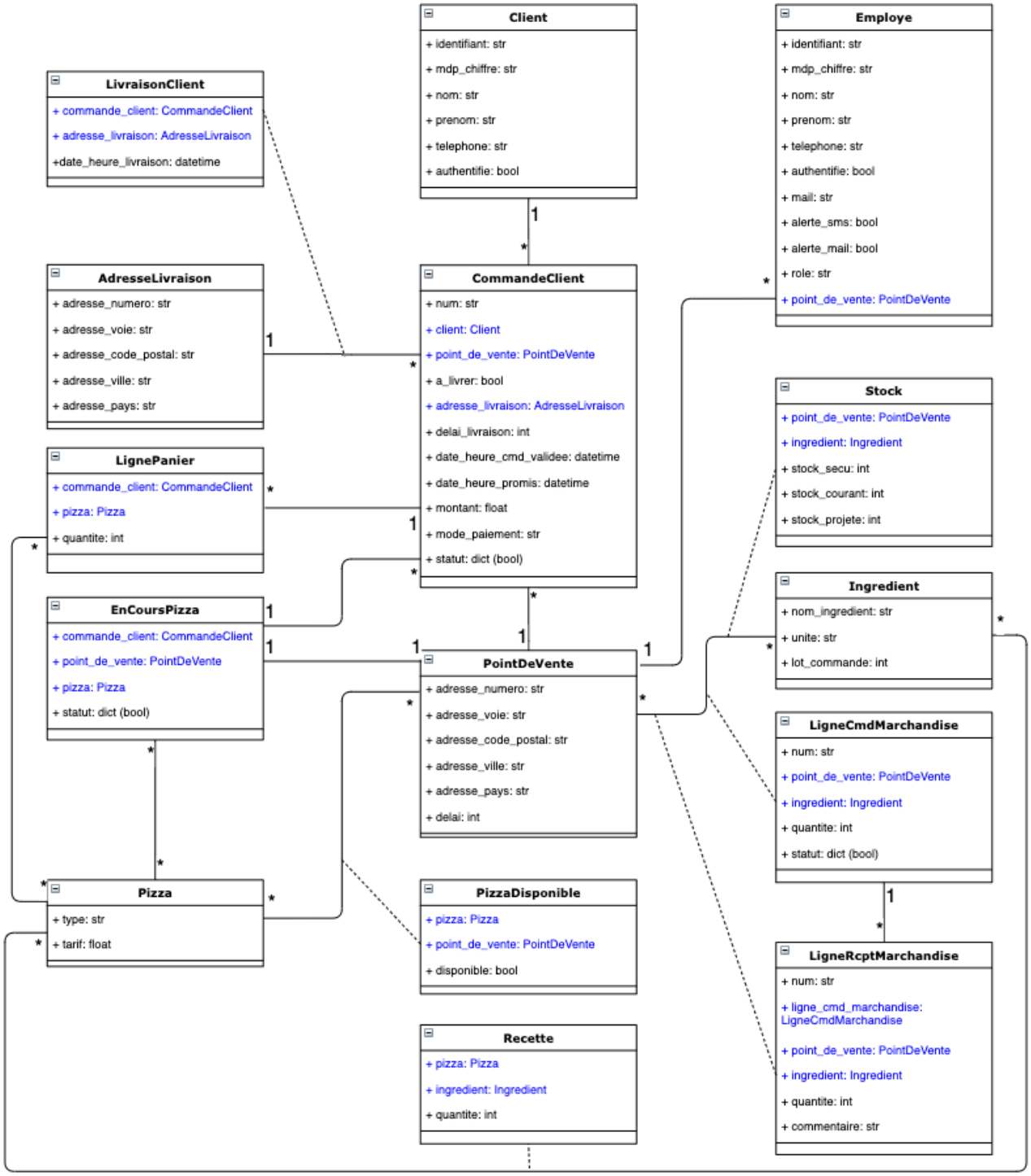

Diagramme de classes

3.2 - Règles de gestion

Sur ce projet, il a été choisi de centrer le diagramme sur les classes *CommandeClient* et *PointDeVente*, comme le montre la figure précédente.

En effet, les orientations prises pendant la rédaction des spécifications fonctionnelles ont guidé cette décision, notamment pour les raisons suivantes :

- la commande du client est l'élément central qui lie l'acteur *client* et les acteurs du *point de vente*
- la sélection du point de vente est effectuée, de façon explicite (choix direct du client) ou implicite (livraison à une adresse fournie par le client), au début du processus de commande

De plus, voici pour rappel les règles de gestion générales s'appliquant au projet, qui ont été présentées à la section 5.3 du dossier de conception fonctionnelle de l'application :

- il n'y a qu'une taille de pizza disponible pour les clients
- la page d'accueil du site web demande de sélectionner le point de vente (dans le cas d'un retrait de commande sur place) ou de saisir l'adresse de livraison (dans le cas contraire)
- une commande contient une ou plusieurs pizzas
- il y a un surcoût tarifaire pour la livraison (par rapport à un retrait en point de vente)
- il n'y a pas de possibilité pour le client de choisir l'heure de mise à disposition de la commande : elle est mise en file d'attente du point de vente selon la règle PEPS (Premier Entré Premier Sorti) et le client est averti de l'heure de disponibilité
- le client a la possibilité d'annuler ou modifier sa commande seulement si celle-ci n'est pas encore en préparation

EB Consulting 1 rue du Pont, 75000 Paris – 01.02.03.04.05 – contact@eb-consulting.com
www.eb-consulting.com S.A.R.L. au capital de 1 000,00 € enregistrée au RCS de Paris – SIREN 999 999 999 – Code APE : 6202A

4 - ARCHITECTURE TECHNIQUE

4.1 - Application Web

La pile logicielle est la suivante :

- Application **Python** (version 3.6.9 ou ultérieure) avec le framework **Django** (version 3.0.5 ou ultérieure)
- Serveur d'application **Gunicorn** (version 20.0.4 ou ultérieure)
- Serveur web **NGINX** (version 1.17.10 ou ultérieure)
- Base de données **PostgreSQL** (version 12.2 ou ultérieure)

Voici le diagramme de composants de l'application « ERP OC Pizza » :

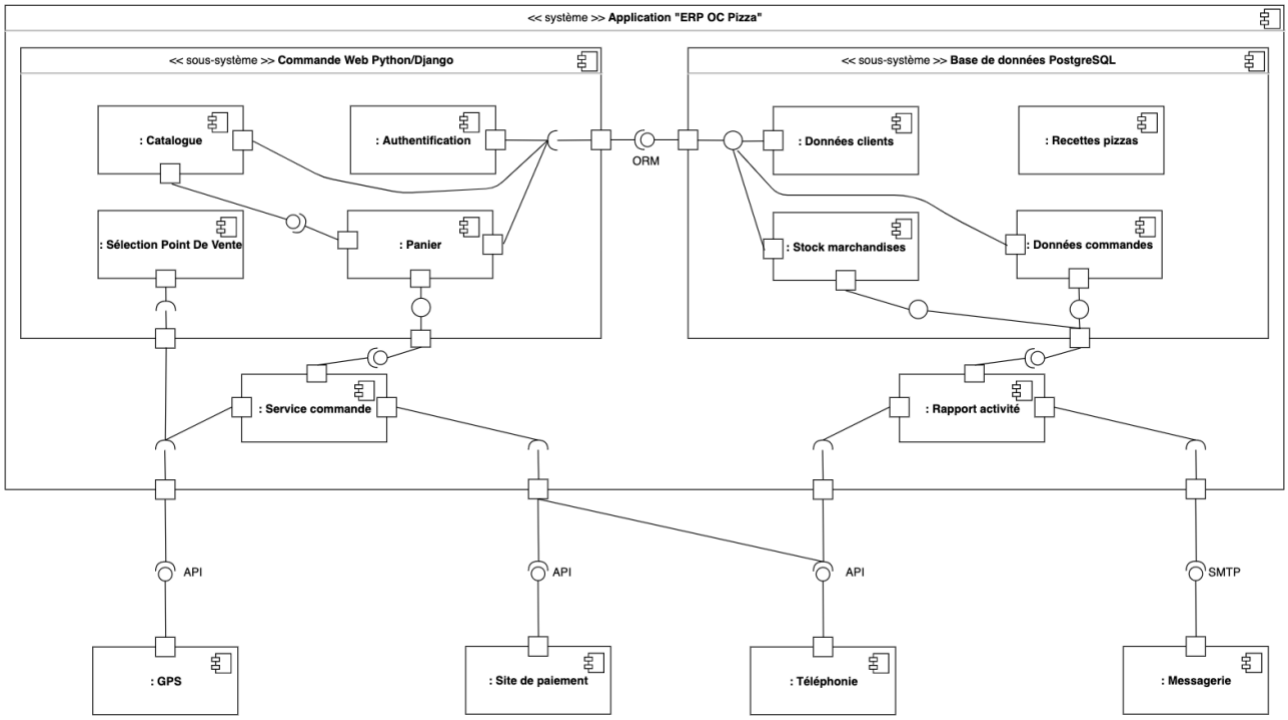

Diagramme de composants

Les composants sont présentés plus en détails dans les sections ci-après.

EB Consulting 1 rue du Pont, 75000 Paris – 01.02.03.04.05 – contact@eb-consulting.com
www.eb-consulting.com S.A.R.L. au capital de 1 000,00 € enregistrée au RCS de Paris – SIREN 999 999 999 – Code APE : 6202A

4.1.1 - Sous-système « Commande web Python/Django »

4.1.1.1 - Composant « Catalogue »

Le *Catalogue* contient la liste des pizzas disponibles à la vente.

Il requiert une interface avec le *Stock marchandises* (du sous-système « Base de données PostgreSQL ») afin de tenir compte d'éventuelles ruptures de stock d'ingrédients.

Il propose une interface avec le *Panier*.

4.1.1.2 - Composant « Authentification »

L'*Authentification* permet de connecter un utilisateur (client ou employé) à l'application web.

Dans le cas d'un client, elle requiert une interface avec les *Données clients* (du sous-système « Base de données PostgreSQL »).

4.1.1.3 - Composant « Sélection Point de Vente »

La *Sélection Point de Vente* est manuelle dans le cas où le client choisit d'aller chercher sa commande sur place, alors qu'elle est automatique dans le cas d'une livraison de la commande.

Dans le cas d'une livraison, elle requiert une interface avec le composant externe *GPS*.

4.1.1.4 - Composant « Panier »

Le *Panier* constitue le contenu d'une commande.

Il requiert une interface avec le *Catalogue* et une autre interface avec les *Données commandes* (du sous-système « Base de données PostgreSQL »).

Il propose une interface avec le *Service commande*.

4.1.2 - Sous-système « Base de données PostgreSQL »

4.1.2.1 - Composant « Données clients »

Les *Données clients* contiennent les informations relatives aux comptes créés par les clients dans l'application.

Il propose une interface avec l'*Authentification* (du sous-système « Commande web Python/Django »).

4.1.2.2 - Composant « Recettes pizzas »

Les *Recettes pizzas* servent d'aide-mémoire aux pizzaïolos pendant la préparation des pizzas.

4.1.2.3 - Composant « Stock marchandises »

Le *Stock marchandises* gère l'inventaire des ingrédients nécessaires à la préparation des pizzas.

Il propose une interface avec le *Catalogue* (du sous-système « Commande web Python/Django ») et une autre interface avec le *Rapport activité*.

4.1.2.4 - Composant « Données commandes »

Les *Données commandes* contiennent à la fois des informations relatives au contenu des commandes, mais aussi à leur statut (en attente, en préparation, soldée, etc.).

Il propose une interface avec le *Panier* (du sous-système « Commande web Python/Django ») et une autre interface avec le *Rapport activité*.

EB Consulting 1 rue du Pont, 75000 Paris – 01.02.03.04.05 – contact@eb-consulting.com
www.eb-consulting.com S.A.R.L. au capital de 1 000,00 € enregistrée au RCS de Paris – SIREN 999 999 999 – Code APE : 6202A

4.1.3 - Autres composants internes

4.1.3.1 - Composant « Service commande »

Le *Service commande* contient les informations relatives au type de service à fournir au client (livraison ou service sur place) ainsi que le statut du paiement (commande déjà payée ou non).

Il requiert une interface avec le *Panier* (du sous-système « Commande web Python/Django »), une autre interface avec le *Site de paiement* (qui est un composant externe) et une dernière interface avec le *GPS* (qui est aussi un composant externe) dans le cas d'une commande à livrer.

4.1.3.2 - Composant « Rapport activité »

Le *Rapport activité* est destiné au gérant et lui fournit des indicateurs en lien avec l'activité du groupe.

Il requiert deux interfaces avec respectivement le *Stock marchandises* et les *Données commandes* (du sous-système « Commande web Python/Django »), une autre interface avec la *Messagerie* (qui est un composant externe) et une dernière interface avec la *Téléphonie* (qui est aussi un composant externe) dans le cas d'une commande à livrer.

4.1.4 - Composants externes

4.1.4.1 - Composant « GPS »

Le *GPS* est utilisé uniquement dans le cas d'une commande à livrer. C'est un service permettant de définir le point de vente de référence (qui jouera alors le rôle de lieu de préparation de la commande), ainsi que de guider le livreur lors de sa tournée.

Il propose une interface avec la *Sélection Point de Vente* (du sous-système « Commande web Python/Django ») et le *Service commande* (qui est un composant interne).

4.1.4.2 - Composant « Site de paiement »

Le *Site de paiement* est utilisé uniquement dans le cas d'un paiement réalisé en ligne depuis l'application.

Il propose une interface avec le *Service commande* (qui est un composant interne).

4.1.4.3 - Composant « Messagerie »

La *Messagerie* est utilisée uniquement par le gérant pour recevoir des alertes par courriel.

Il propose deux interfaces avec respectivement le *Rapport activité* et le Service commande (qui sont des composants internes).

4.1.4.4 - Composant « Téléphonie »

La *Téléphonie* est utilisée par le gérant pour recevoir des alertes par SMS, mais aussi par le livreur pour connaître l'adresse et le contenu d'une commande mentionnés dans un SMS.

Il propose une interface avec le *Rapport activité* (qui est un composant interne).

EB Consulting 1 rue du Pont, 75000 Paris – 01.02.03.04.05 – contact@eb-consulting.com
www.eb-consulting.com S.A.R.L. au capital de 1 000,00 € enregistrée au RCS de Paris – SIREN 999 999 999 – Code APE : 6202A

5 - Architecture de Deploiement

Voici le diagramme de déploiement du projet :

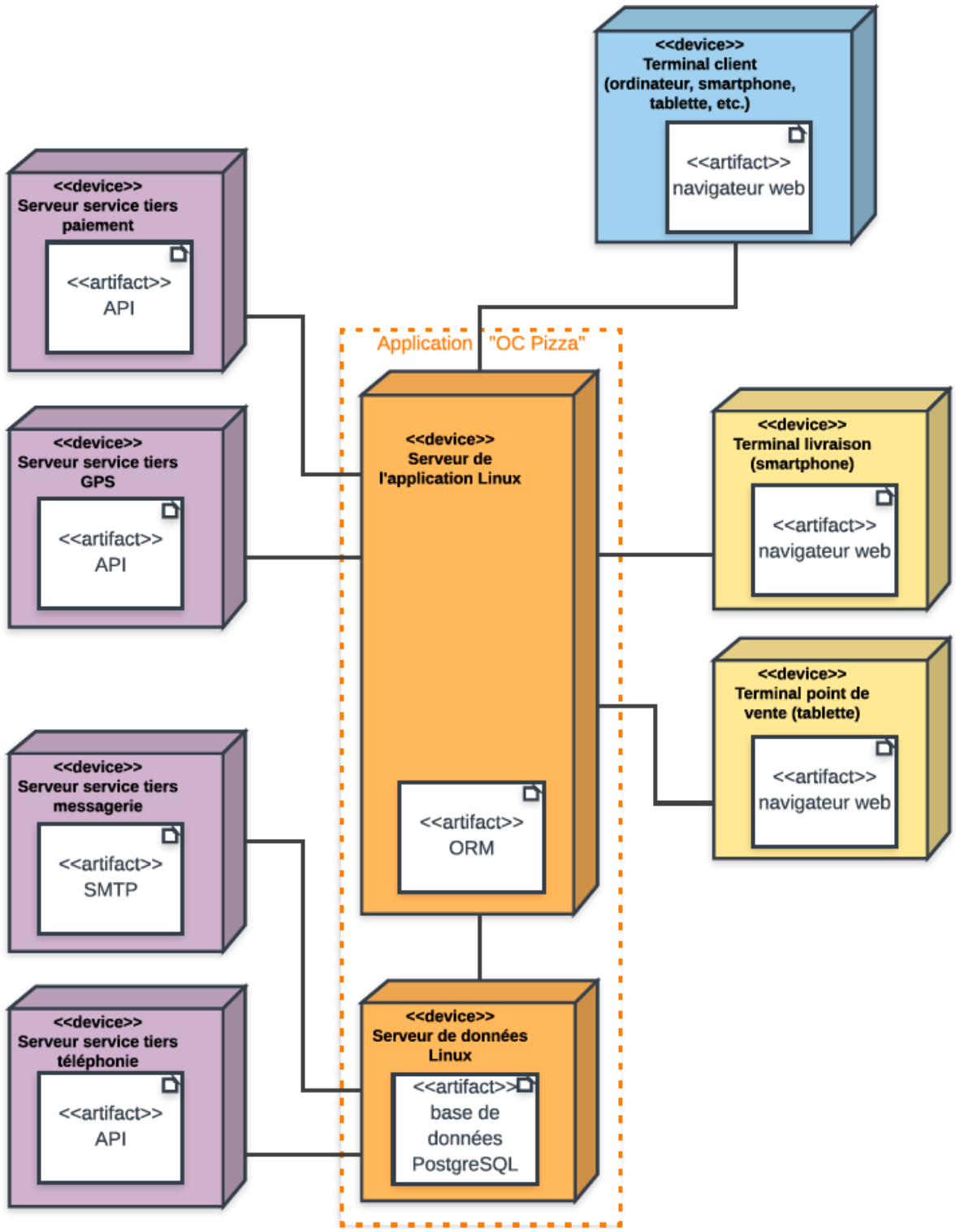

Diagramme de déploiement

EB Consulting 1 rue du Pont, 75000 Paris – 01.02.03.04.05 – contact@eb-consulting.com
www.eb-consulting.com S.A.R.L. au capital de 1 000,00 € enregistrée au RCS de Paris – SIREN 999 999 999 – Code APE : 6202A

La partie centrale de ce diagramme de déploiement (en orange) représente l'application « ERP OC Pizza » et elle est installée sur deux entités physiques :

- le serveur de l'application sur lequel sont notamment installés l'application Python/Django et le serveur web
- un serveur de données sur lequel est installée la base de données

Les quatre blocs à gauche (en violet) représentent les services tiers nécessaires au fonctionnement de l'application.

Le bloc situé dans la partie supérieure du diagramme de déploiement (en bleu) représente l'appareil utilisé par le client pour interagir avec l'application web.

Enfin, les deux blocs situés à droite (en jaune) représentent les terminaux utilisés par les employés de la société OC Pizza : le smartphone pour le livreur et la tablette pour le pizzaïolo.

5.1 - Serveur de base de données

Le serveur de base de données héberge les données nécessaires au fonctionnement de l'application. Son rôle consiste à stocker les données et à les transmettre au serveur d'application.

Le serveur utilisé ici est un serveur Linux Ubuntu (version 18.04 LTS ou LTS ultérieure) sur lequel est installé PostgreSQL (version 12.2 ou ultérieure).

La structure de la base de données est indiquée dans le Modèle Physique de Données présenté en annexe : **PDOCPizza_02B_annexe_MPD**.

5.2 - Serveur d'application

Le serveur d'application est le serveur principal de ce projet.

Le serveur utilisé ici est un serveur Linux Ubuntu (version 18.04 LTS ou LTS ultérieure) sur lequel sont installés :

- Python (version 3.6.9 ou ultérieure), le langage de programmation back end du projet
- Gunicorn (version 20.0.4 ou ultérieure), qui traite les requêtes HTTP et les transmet au code Python/Django
- Django (version 3.0.5 ou ultérieure), le framework web Python utilisé pour ce projet
- les autres dépendances de Python nécessaires au projet

Gunicorn pilote l'exécution de l'application web.

5.3 - Serveur web

Le serveur web est l'interface entre le navigateur web de l'utilisateur et le serveur d'application. Il a pour rôle de rediriger le flux entrant.

Ici, le serveur web utilisé est physiquement le serveur d'application décrit dans la section 5.2. Le logiciel installé est NGINX (version 1.17.10 ou ultérieure). Celui-ci sert directement les fichiers statiques, et redirige vers le serveur d'application Gunicorn le reste du flux entrant.

EB Consulting 1 rue du Pont, 75000 Paris – 01.02.03.04.05 – contact@eb-consulting.com
www.eb-consulting.com S.A.R.L. au capital de 1 000,00 € enregistrée au RCS de Paris – SIREN 999 999 999 – Code APE : 6202A

6 - ARCHITECTURE LOGICIELLE

6.1 - Principes généraux

Les sources et versions du projet sont gérées par Git.

Les dépendances sont gérées par PyPI (Python Package Index), elles sont référencées dans le document *requirements.txt* et elles sont installées dans un environnement virtuel géré par venv.

6.1.1 - Les applications Django

Les applications Django sont des subdivisions du projet ayant des responsabilités différentes. Les applications Django nécessaires à ce projet sont les suivantes :

- authentification de l'utilisateur : *custom_auth*
- consultation et/ou modification d'un compte utilisateur : *account*
- consultation du catalogue de produits : *search*
- commande d'une ou plusieurs pizzas : *purchase*
- préparation des commandes : *cook*
- service des commandes : *delivery*
- gestion des stocks : *inventory*
- pilotage de l'activité : *monitoring*

6.1.2 - Structure des sources

La structuration des répertoires du projet suit l'arborescence par défaut du framework Django :

```
erp_oc_pizza/
├─ erp_oc_pizza/
│   ├─ __init__.py
│   ├─ settings/
│   │   ├─ __init__.py
│   │   ├─ production.py
│   │   ├─ travis.py
│   ├─ asgi.py
│   ├─ urls.py
│   ├─ wsgi.py
├─ account/
├─ custom_auth/
├─ cook/
├─ delivery/
├─ inventory/
├─ monitoring/
├─ purchase/
├─ search/
├─ staticfiles/
├─ templates/
├─ tests/
├─ favicon.png
├─ manage.py
├─ Procfile
├─ requirements.txt
├─ runtests.py
```

Chaque **application Django** contient plus ou moins la même arborescence. En effet, en fonction des besoins, des modules peuvent être ajoutés ou retirés dans ces packages. Cependant, le socle commun à chacun de ces packages est le suivant :

```
application_django/
├─ __init__.py
├─ migrations/
├─ templates/
│   ├─ application_django/
├─ admin.py
├─ apps.py
├─ forms.py
├─ models.py
├─ urls.py
├─ views.py
```

staticfiles/ → contient les fichiers statiques du projet

templates/ → contient les templates d'erreur : 404.html et 500.html

tests/ → contient les tests unitaires, les tests fonctionnels et les tests d'intégration

EB Consulting 1 rue du Pont, 75000 Paris – 01.02.03.04.05 – contact@eb-consulting.com
www.eb-consulting.com S.A.R.L. au capital de 1 000,00 € enregistrée au RCS de Paris – SIREN 999 999 999 – Code APE : 6202A

7 - GLOSSAIRE

Fichiers statiques	Ce sont les fichiers qui ne sont jamais modifiés par l'exécution du code côté serveur. Les exemples les plus courants sont les fichier CSS (extension « .css ») et Javascript (extension « .js »).
Module	Dans le cadre d'une arborescence Python, il s'agit d'un fichier source codé en Python et portant l'extension « .py ».
Package	Dans le cadre d'une arborescence Python, il s'agit d'un répertoire contenant un ou plusieurs modules (au minimum le module *__init__.py*).
Template	Également appelé « gabarit ». Dans le cadre des fichiers HTML, c'est le document qui est affiché par le navigateur web. Il peut faire appel à des fichiers statiques, et inclure des données statiques et dynamiques.

EB Consulting 1 rue du Pont, 75000 Paris – 01.02.03.04.05 – contact@eb-consulting.com
www.eb-consulting.com S.A.R.L. au capital de 1 000,00 € enregistrée au RCS de Paris – SIREN 999 999 999 – Code APE : 6202A