



# OC Pizza

## Projet 9

Dossier de conception technique

Version 1.1

**Auteur**

Emmanuel Nocquet  
*Développeur web*

# TABLE DES MATIÈRES

<b>1 - Versions.....</b>	<b>3</b>
<b>2 - Introduction.....</b>	<b>4</b>
2.1 - Objet du document.....	4
2.2 - Références.....	4
<b>3 - Architecture Technique.....</b>	<b>5</b>
3.1 - Application Web.....	5
3.2 - Base de données.....	6
3.2.1 - L'architecture choisie.....	6
3.2.2 - Le modèle physique de données.....	6
<b>4 - Architecture de Déploiement.....</b>	<b>8</b>
4.1 - Serveur de déploiement.....	8
4.2 - Les composants du déploiement.....	8
<b>5 - Architecture logicielle.....</b>	<b>10</b>
5.1 - Principes généraux.....	10
5.1.1 - Les applications.....	10
5.1.2 - Structure des sources.....	10
<b>6 - Points particuliers.....</b>	<b>12</b>
6.1 - Gestion des logs.....	12
6.2 - Ressources.....	12
6.3 - Environnement de développement.....	12
6.4 - Procédure de packaging / livraison.....	12
<b>7 - Glossaire.....</b>	<b>13</b>

# 1 - VERSIONS

Auteur	Date	Description	Version
E. Nocquet	04/02/2020	Création du document	1.0
E. Nocquet	02/05/2020	Ajout éléments avant livraison client	1.1

## 2 - INTRODUCTION

### 2.1 - Objet du document

Le présent document constitue le dossier de conception technique de l'application "oc\_pizza"

L'objectif du document est de décrire les solutions techniques choisies pour le développement de l'application "oc\_pizza".

Les éléments du présents dossiers découlent :

- de l'expression des besoins du client
- de notre analyse sur la faisabilité en tant qu'expert du développement web

### 2.2 - Références

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

1. **DCF - 002** : Dossier de conception fonctionnelle de l'application
2. **DE - 003**: Dossier d'exploitation de l'application

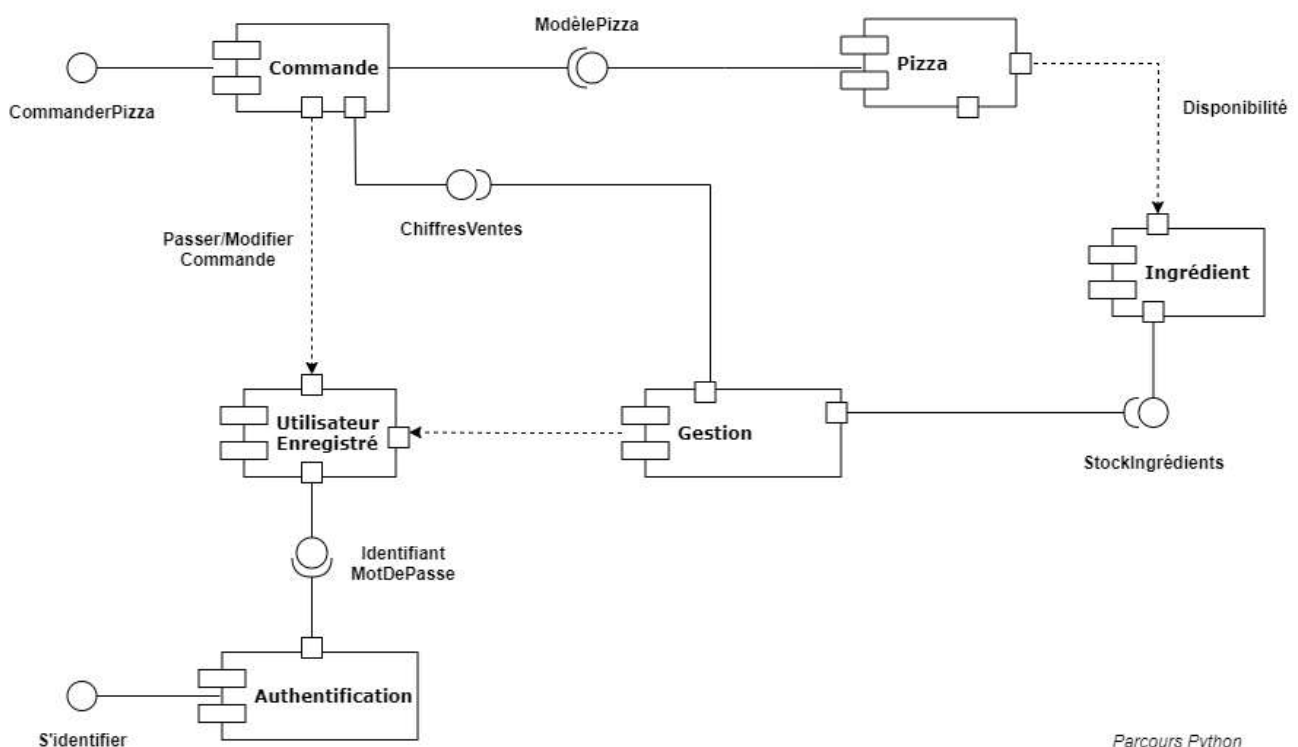
# 3 - ARCHITECTURE TECHNIQUE

## 3.1 - Application Web

L'application sera écrite en Python (version 3.8) avec le framework Django (version 3.0).

Pour décrire les éléments du système, on utilise un diagramme de composants: celui-ci décrit le système comme un ensemble de composants réutilisables mettant en évidence leurs relations de dépendance.

### Diagramme de composant



Parcours Python  
Projet 06  
Emmanuel Nocquet

Dans ce diagramme les composants “commande”, “pizza” ou “gestion” fournissent des services précis: ils englobent des fonctionnalités du système cohérentes entre elles.

Commande ou Pizza, Gestion ou Authentification sont des éléments qui ont vocation à être réutilisés pour accéder à des fonctionnalités précises du futur logiciel : le composant Commande englobe les fonctionnalités liées aux commandes passées et tout ce que cela suppose, la gestion de la préparation des pizzas, leurs livraisons, les tarifs, alors que le composant Pizza sera dédié aux recettes existantes dans le restaurant, la composition des pizzas, les disponibilités, les tarifs...

Un composant "gestion" spécifiquement dédié à l'usage du manager du restaurant dans le cadre de sa mission de gestion globalisée du restaurant, apparaît également. Comme cela a été demandé par le client, ce composant permettra d'avoir accès aux informations fournies par les composants commande et ingrédient.

## 3.2 - Base de données

### 3.2.1 - *L'architecture choisie*

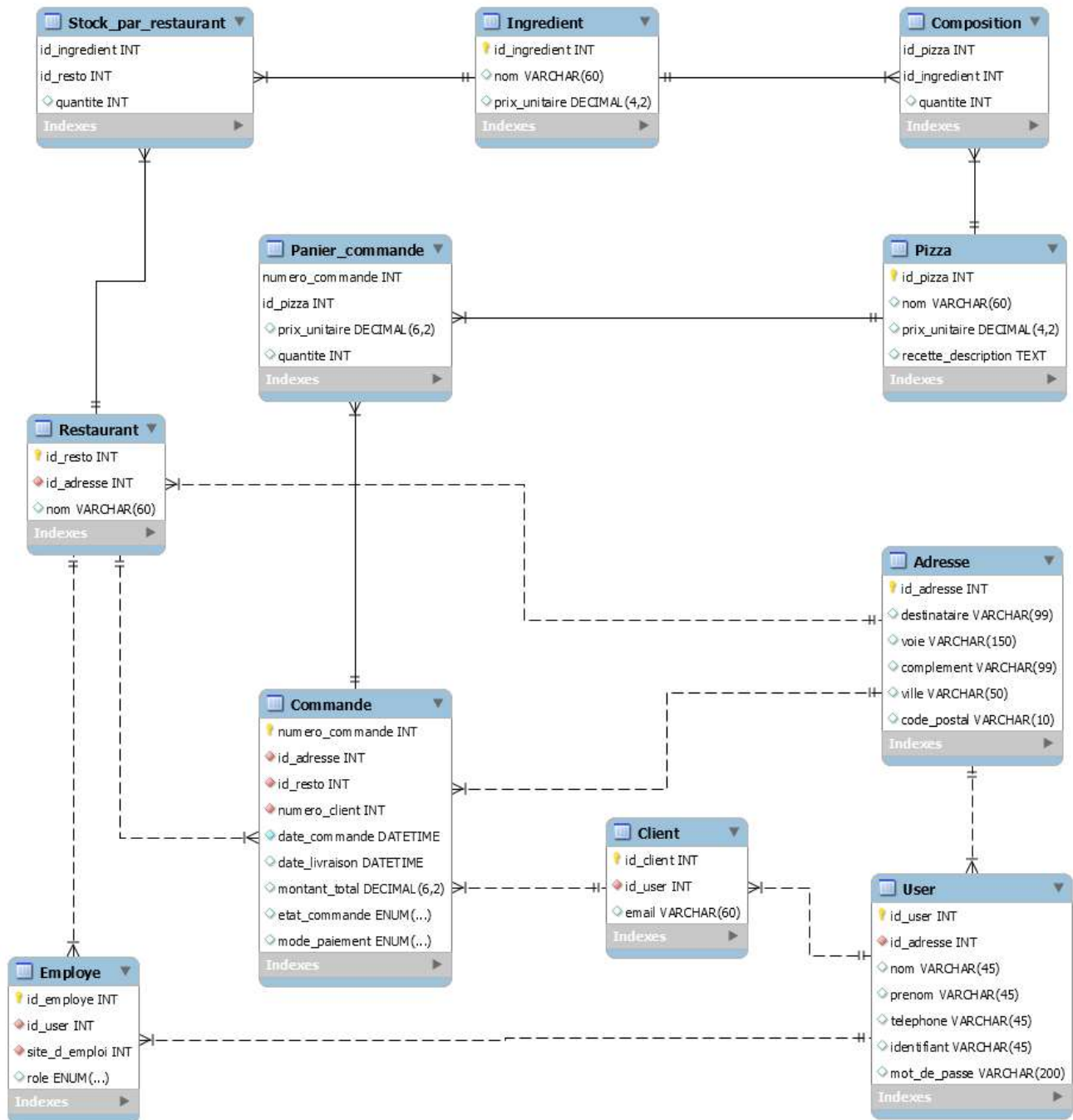
L'application utilisera une base de données PostgreSQL: il s'agit d'un des système de gestion de base de données les plus complets et, il est compatible avec le déploiement de l'application sur la plateforme "Heroku".

### 3.2.2 - *Le modèle physique de données*

Il s'agit de l'aboutissement de notre travail. Issu de toutes les études précédentes, le modèle physique de données représente le format de la base de données telle qu'elle pourrait être implementée dans la future application "OC\_Pizza".

Dans le diagramme suivant, les différentes tables stockent les informations en rapport avec les entités identifiées lors des étapes successives de notre étude, au format qui convient pour ces données.

OCPizza - Modèle physique de données



**Modèle physique de données (oc\_pizza)**

# 4 - ARCHITECTURE DE DÉPLOIEMENT

## 4.1 - Serveur de déploiement

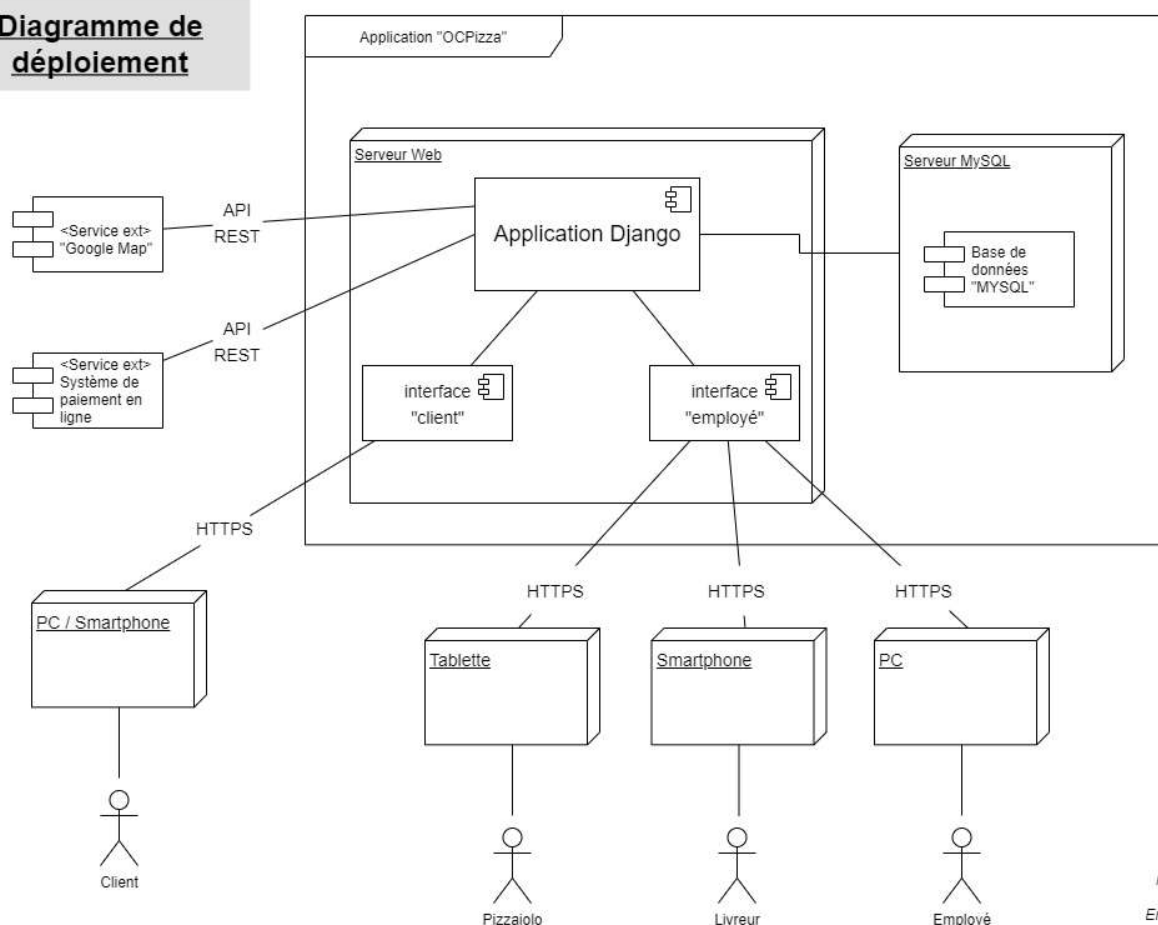
L'application "oc\_pizza" sera déployée sur 'Heroku': il s'agit d'une plateforme web de type 'PaaS' (platform as a service) qui permet aux développeurs d'héberger des applications sur le cloud.

Cette solution semble adaptée aux dimensions du projet et facilitera les opérations de déploiement et de maintenance.

## 4.2 - Les composants du déploiement

Pour représenter l'utilisation de l'infrastructure physique (les serveurs, ordinateurs...) par le système, et la manière dont les composants sont répartis ainsi que leurs relations entre eux, on utilise un diagramme de déploiement.

### Diagramme de déploiement



Parcours Python  
Projet 06  
Emmanuel Nocquet



Ce diagramme fait le lien entre l'application et le monde "physique" avec la présence des acteurs amenés à interagir avec l'application. Ces acteurs peuvent interagir en fonction de leur qualité (client/employé) et de leur profil dans ce dernier cas.

L'application est accessible via internet, elle est responsive et s'adapte au type d'écran utilisé: par exemple, le pizzaiolo se voit spécifiquement attribué une interface de type tablette pour consulter les commandes en cours et les recettes de pizzas; l'accès depuis un smartphone sera également possible, ce qui sera notamment utile pour le livreur afin qu'il mette à jour le statut de la commande.

Cette distinction entre employé et client se retrouve au niveau des composants de l'application, eux mêmes liés à l'application Django. Cette application Django est le coeur du logiciel (le back end), elle interagit avec tous les éléments (base de donnée MySQL, service API paiement/GoogleMap...) que l'on retrouve sur ce diagramme.

# 5 - ARCHITECTURE LOGICIELLE

## 5.1 - Principes généraux

Les sources et versions du projet sont gérées par **Git**, les dépendances par le fichier **requirements.txt**, le fichier **procfile** indique à la plateforme qu'il s'agit d'une application web.

La structure d'un projet 'Django' suppose la création d'un projet puis d'une ou plusieurs applications amenées à interagir au sein du projet.

### 5.1.1 - Les applications

L'architecture applicative est la suivante :

- L'application **oc\_pizza** : responsable de la logique de l'application web. C'est autour d'elle que s'articulent les autres éléments
- une application **commercial** : implémentation de l'offre de vente de pizzas en ligne et des éléments s'y rapportant
- une application **gestion\_interne** : implémentation des fonctions de gestion destinées aux managers du groupe

### 5.1.2 - Structure des sources

La structuration des répertoires du projet suit la logique suivante :

```
racine
├── procfile
├── oc_pizza
│   ├── __init__.py
│   ├── settings.py
│   ├── urls.py
│   ├── manage.py
│   └── wsgi.py
├── commercial
│   ├── __init__.py
│   ├── urls.py
│   ├── admin.py
│   ├── views.py
│   ├── models.py
│   └── apps.py
├── gestion_interne
│   ├── __init__.py
│   ├── urls.py
│   ├── admin.py
│   ├── views.py
│   ├── models.py
│   └── apps.py
├── README.md
├── staticfiles
└── requirements.txt
```

Comme définit lors de l'étude fonctionnelle, les fonctionnalités de l'application web « oc\_pizza » apparaissent dans des applications dédiées soit à la partie commerciale du site, soit à la partie 'gestion interne'.

## 6 - POINTS PARTICULIERS

### 6.1 - Gestion des logs

Comme indiqué dans le point 7 du dossier d'exploitation de l'application "oc\_pizza", la gestion des fichiers de journalisation (logs) est du ressort du client une fois l'application déployée. Néanmoins, il convient de mener une réflexion en amont sur le sujet, et IT Consulting & Development peut accompagner le client dans ce domaine également.

### 6.2 - Ressources

L'équipe de développement de IT Consulting & Development s'est appuyé sur les ressources fournies par le client, notamment en ce qui concerne la charte graphique, mais également tout ce qui concerne les données de travail du client (recettes, tarifs, clients ...)

### 6.3 - Environnement de développement

L'équipe de développement de IT Consulting & Development a utilisé dans le cadre de ce projet l'IDE Visual Studio de Microsoft.

### 6.4 - Procédure de packaging / livraison

L'application "oc\_pizza" sera déployée sur les serveurs de la plateforme "Heroku" avant la livraison au client. IT Consulting & Development fournira toute la documentation nécessaire à ce moment-là.

## 7 - GLOSSAIRE

<b>Framework</b>	Un framework désigne un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel.
<b>Python</b>	Python est un langage de programmation interprété, multi-paradigme et multiplateformes.
<b>Django</b>	Django est un cadre de développement web (framework) open source en Python.
<b>API</b>	En informatique, une interface de programmation d'application ou interface de programmation applicative est un ensemble normalisé de classes, de méthodes, de fonctions et de constantes qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels.
<b>Git</b>	Git est un logiciel de gestion de versions décentralisé.
<b>Responsive</b>	Un site web responsive est un site web dont la conception vise, grâce à différents principes et techniques, à offrir une consultation confortable sur des écrans de tailles très différentes.
<b>PaaS</b>	Plate-forme en tant que service, PaaS, de l'anglais platform as a service, est l'un des types de cloud computing, principalement destiné aux développeurs ou aux entreprises de développement.
<b>IDE</b>	En programmation informatique, un environnement de développement est un ensemble d'outils qui permet d'augmenter la productivité des programmeurs qui développent des logiciels.