



# OC Pizza

## Application web - Site de commerce et de logistique

Dossier de conception technique

Version 1.1

**Auteur**

Kilian Florin

*Analyste-Programmeur*





## TABLE DES MATIERES

<b>1 - Versions.....</b>	<b>4</b>
<b>2 - Introduction .....</b>	<b>5</b>
2.1 - Objet du document.....	5
2.2 - Références.....	5
<b>3 - Architecture Technique.....</b>	<b>6</b>
3.1 - Application Web .....	6
<b>4 - Architecture de Déploiement .....</b>	<b>7</b>
4.1 - Serveur de Base de données.....	7
4.2 - Serveur d'Application Web.....	7
<b>5 - Architecture logicielle.....</b>	<b>8</b>
5.1 - Principes généraux .....	8
5.1.1 - Les couches.....	8
5.1.2 - Les modules.....	8
5.1.3 - Structure des sources .....	8
5.2 - Application Web .....	9
<b>6 - Points particuliers.....</b>	<b>10</b>
6.1 - Gestion des logs .....	10
6.2 - Fichiers de configuration .....	10
6.2.1 - Application web.....	10
6.2.1.1 - Fichier Production.py .....	10
6.2.1.2 - Fichier ocpizza (Nginx) .....	10
6.2.1.3 - Fichier ocpizza-gunicorn.conf (Supervisor) .....	10
6.3 - Environnement de développement .....	10
6.3.1 - Virtual Env.....	10
<b>7 - Glossaire.....</b>	<b>11</b>



# 1 - VERSIONS

Auteur	Date	Description	Version
Kilian Florin	03/04/2018	Création du document	1.0
Kilian Florin	26/04/2018	Mise à jour des informations global	1.1



## 2 - INTRODUCTION

### 2.1 - Objet du document

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

Les objectifs du document sont d'apporter des précisions sur les choix techniques, technologique et architecturaux du projet.

### 2.2 - Références

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

1. **DCF – OC Pizza** : Dossier de conception fonctionnelle de l'application
2. **DE – OC Pizza** : Dossier d'exploitation de l'application



## 3 - ARCHITECTURE TECHNIQUE

### 3.1 - Application Web

La pile logicielle est la suivante :

- **Python** version 3.5.0 ou +
- **Virtualenv** : version 15.0.0
- **Supervisor** : version 3.2.0-2 ou +
- **Nginx** : version 1.9.15 ou +
- **Git** : version 1:2.7.4 ou +
- **PostgreSQL** : version 10



## 4 - ARCHITECTURE DE DÉPLOIEMENT



### 4.1 - Serveur de Base de données

Serveur de base de données si nécessaire d'être séparé du serveur d'application web.

Caractéristiques techniques :

OS : Linux Ubuntu Xenial Xerus (version 16.x.x)

PostgreSQL : version 10

### 4.2 - Serveur d'Application Web

OS : Linux Ubuntu Xenial Xerus (version 16.x.x)



## 5 - ARCHITECTURE LOGICIELLE

### 5.1 - Principes généraux

Les sources et versions du projet sont gérées par **Git** et les dépendances par **PyPi**

#### 5.1.1 - Les couches

L'architecture applicative est la suivante :

- Une couche **model** : implémentation du modèle des objets métiers
- Une couche **template** : représente les gabarits des données renvoyer
- Une couche **vue** : donne accès au modèle et au template adapté à la requête

#### 5.1.2 - Les modules

Django-Cron : Permet de créer des tâches « Cron » pouvant s'exécuter à intermédiaire régulier, comme une mise à jour de la base de données vis-à-vis d'une source externe.

#### 5.1.3 - Structure des sources

La structuration des répertoires du projet suit la logique suivante :

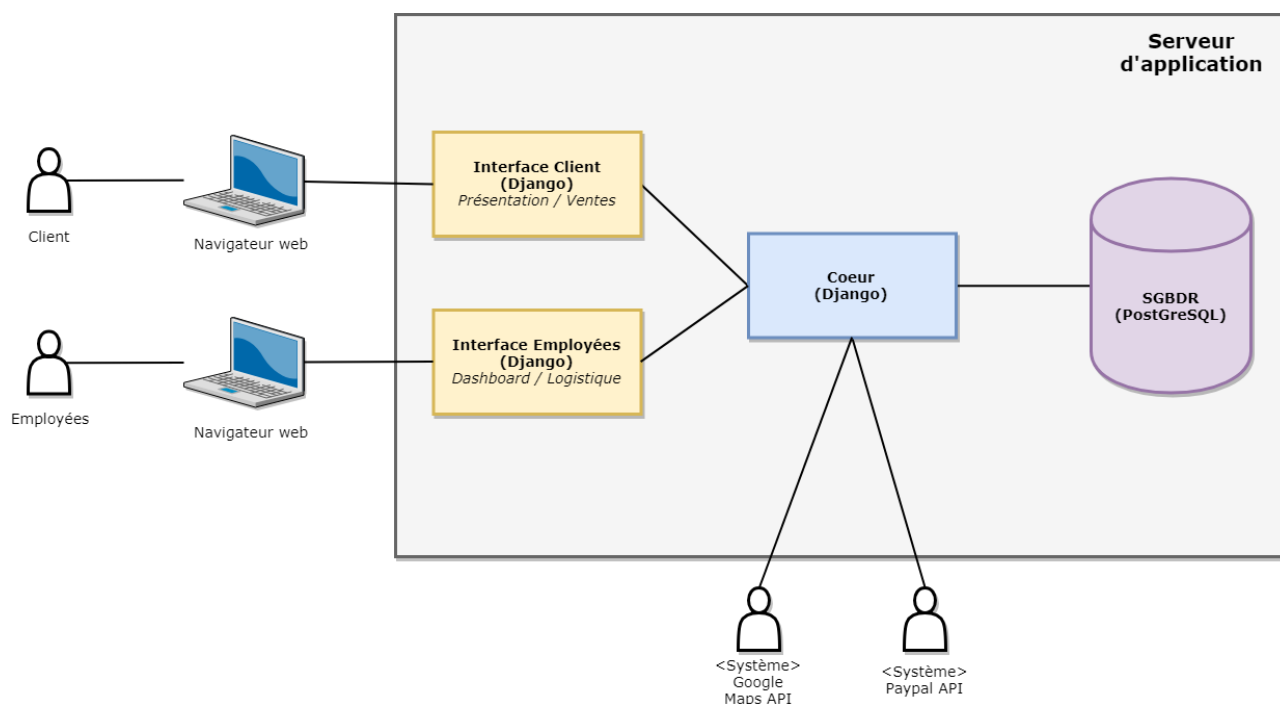
- Les répertoires sources sont créés automatiquement par Django sur lesquels nous ajoutons le dossier staticfiles ainsi que l'arborescence pour les templates

```
racine
├── requirements.txt
├── manage.py
├── <project>_project
│   ├── urls.py
│   ├── wsgi.py
│   └── settings
│       └── production.py
│           └── travis.py
├── <applicationX>
│   ├── urls.py
│   ├── models.py
│   ├── views.py
│   ├── tests.py
│   ├── template
│   │   ├── index.html
│   │   └── ...html
│   └── static
│       └── js/css/etc
└── staticfiles
```





## 5.2 - Application Web





## 6 - POINTS PARTICULIERS

### 6.1 - Gestion des logs

Gestion des logs par sentry.io, un lien sera créé lors du déploiement. (Voir Dossier d'exploitation)

### 6.2 - Fichiers de configuration

#### 6.2.1 - Application web

##### 6.2.1.1 - Fichier *Production.py*

**Localisation :** <votre dossier d'installation>/ocpizza/ocpizza\_project/settings/

Il établies les différents modules additionnels au projet tel que le support de Raven qui permet d'envoyer les informations à sentry.io pour le logging ou encore l'authentification à la base de données.

##### 6.2.1.2 - Fichier *ocpizza (Nginx)*

**Localisation :** /etc/nginx/sites-available/ocpizza

Il permet de dicter le comportement de Nginx vis-à-vis du projet en fonction des ressources demandé. Un fichier présente dans l'arborescence « static » de notre projet sera renvoyés directement par Nginx sans passé par notre application Django afin de gagner en performance.

##### 6.2.1.3 - Fichier *ocpizza-gunicorn.conf (Supervisor)*

**Localisation :** /etc/supervisor/conf.d/ocpizza-gunicorn.conf

Il permet de définir les paramètres et le contexte dans lequel Supervisor va exécuter notre application. Il à pour but de maintenir notre application en fonctionnement quoi qu'il arrive dans un environnement de production.

### 6.3 - Environnement de développement

#### 6.3.1 - Virtual Env

Notre application utilise VirtualEnv afin de concentrée l'ensemble des librairies (gérer par Pypi) nécessaire à notre application dans son seul dossier.



## 7 - GLOSSAIRE
