

# OC PIZZA

-

DOSSIER DE CONCEPTION TECHNIQUE

Version 1.0

Auteur  
Raphaël Goupille  
Développeur iOS

# TABLE DES MATIÈRES

<b>1. Versions</b>	
<b>2. Introduction</b>	
2.1. Objet du document .....	4
2.2. Références .....	4
<b>3. Architecture Technique</b>	
3.1. Composants généraux .....	5
3.2. Application Web .....	6
<b>4. Architecture de Déploiement</b>	
4.1. Serveur de Base de données .....	8
4.1.1. <i>Caractéristiques physiques</i> .....	8
4.1.2. <i>Modèle physique de données</i> .....	8
<b>5. Architecture logicielle</b>	
5.1. Principes généraux .....	10
5.1.1. <i>Les couches</i> .....	10
5.1.2. <i>Les modules</i> .....	10
5.1.3. <i>Structure des sources</i> .....	10
<b>6. Points particuliers</b>	
6.1. Gestion des logs .....	13
6.2. Fichiers de configuration .....	13
6.3. Environnement de développement .....	13
6.4. Procédure de packaging / livraison .....	14

# 1. VERSIONS

Auteur	Date	Description	Version
Raphaël Goupille	01/11/2021	Création du document	1.0

## 2.INTRODUCTION

### 2.1.Objet du document

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

L'objet de ce document est de présenter les solutions utilisées pour satisfaire les contraintes d'utilisation de l'application.

Les éléments du présents dossiers découlent des échanges préliminaires ayant eu lieu avec OCPizza au lancement du projet ainsi que des échanges ultérieurs lors de la phase de spécifications des besoins.

### 2.2.Références

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

1. **DCF - 01** : Dossier de conception fonctionnelle de l'application
2. **DEX - 01** : Dossier d'exploitation

### 3.1.Composants généraux

Ces composants sont autonomes et encapsulent un certain nombre de structures complexes.

The diagram illustrates the following components and their interactions:

- Customer** (Boundary) provides a **payment** interface required by **Payment** and a **reading** interface required by **Menu**. It also provides a **Connecting** interface required by **Accounts**.
- Accounts** (Boundary) provides a **Connecting** interface required by **Customer** and an **Account Management** interface required by **Manager**.
- Menu** (Boundary) provides a **reading** interface required by **Customer** and an **ordering** interface required by **Order**.
- Payment** (Boundary) provides a **payment** interface required by **Customer** and an **update** interface required by **Order**.
- Order** (Boundary) provides an **ordering** interface required by **Menu** and an **update** interface required by **Order Tracking**.
- Order Tracking** (Boundary) provides an **update** interface required by **Order** and an **update** interface required by **Employees**.
- Employees** (Boundary) contains three sub-components:
  - Pizzaïolo** (Boundary) provides an **update** interface required by **Order Tracking** and an **update** interface required by **Cooking**.
  - DeliveryMan** (Boundary) provides an **update** interface required by **Order Tracking** and an **update** interface required by **Delivery**.
  - Manager** (Boundary) provides an **update** interface required by **Order Tracking** and an **update** interface required by **Delivery**.
- Stock** (Boundary) provides an **update** interface required by **Order Tracking** and an **update** interface required by **Cooking**.
- Recipe** (Boundary) provides an **update** interface required by **Order Tracking** and a **reading** interface required by **Cooking**.
- Cooking** (Boundary) provides a **reading** interface required by **Recipe** and an **update** interface required by **Pizzaïolo**.
- Delivery** (Boundary) provides an **update** interface required by **DeliveryMan** and an **update** interface required by **Manager**.

## 3.2.Application Web

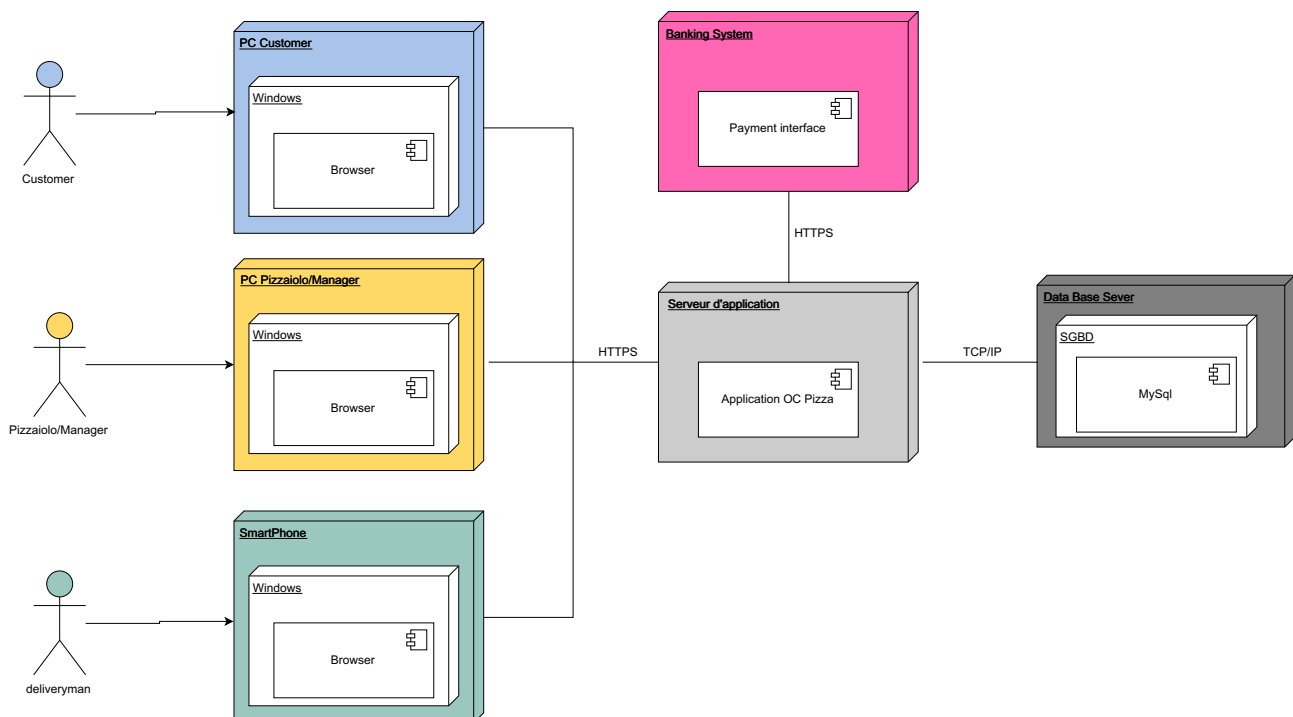
La pile logicielle est la suivante :

- **Linux** (Ubuntu v. 20.04) en tant que système d'exploitation ;
- **Apache** (v. 2.4.46) en guise de serveur Web ;
- **MySQL** (v. 8.0.23) en base de données ;
- **PHP** (v. 8.0.2) en interpréteur de script ; couplé au framework symfony (v. 5.2.4)

Ce sont des logiciels open source, ont une licence dite libre et sont gratuits. Cette pile logicielle est tout à fait adaptée au développement de sites Web dynamiques, et bénéficie d'un support très actif de la communauté.

## 4.ARCHITECTURE DE DÉPLOIEMENT

Le diagramme de déploiement représente la disposition physique des ressources matérielles qui constituent le système et montre la répartition des composants sur ces matériels. Le diagramme montre également les connexions de communication entre les différentes ressources matérielles.



## 4.1. Serveur de Base de données

### 4.1.1. Caractéristiques physiques

Le serveur de **base de données et le serveur web** seront hébergés auprès du prestataire OVH. Les caractéristiques techniques présentées ci-après correspondent à l'offre Rise. L'abonnement est facturé 52,24 € par mois.

- **Processeur:** Intel Xeon E3-1245v5 - 4 c / 8 t - 3.5 GHz / 3.9 GHz
- **Mémoire:** À partir de 32Go
- **Stockage:** NVMe, SATA disponible
- **Bande passante publique:** 500 Mbit/s

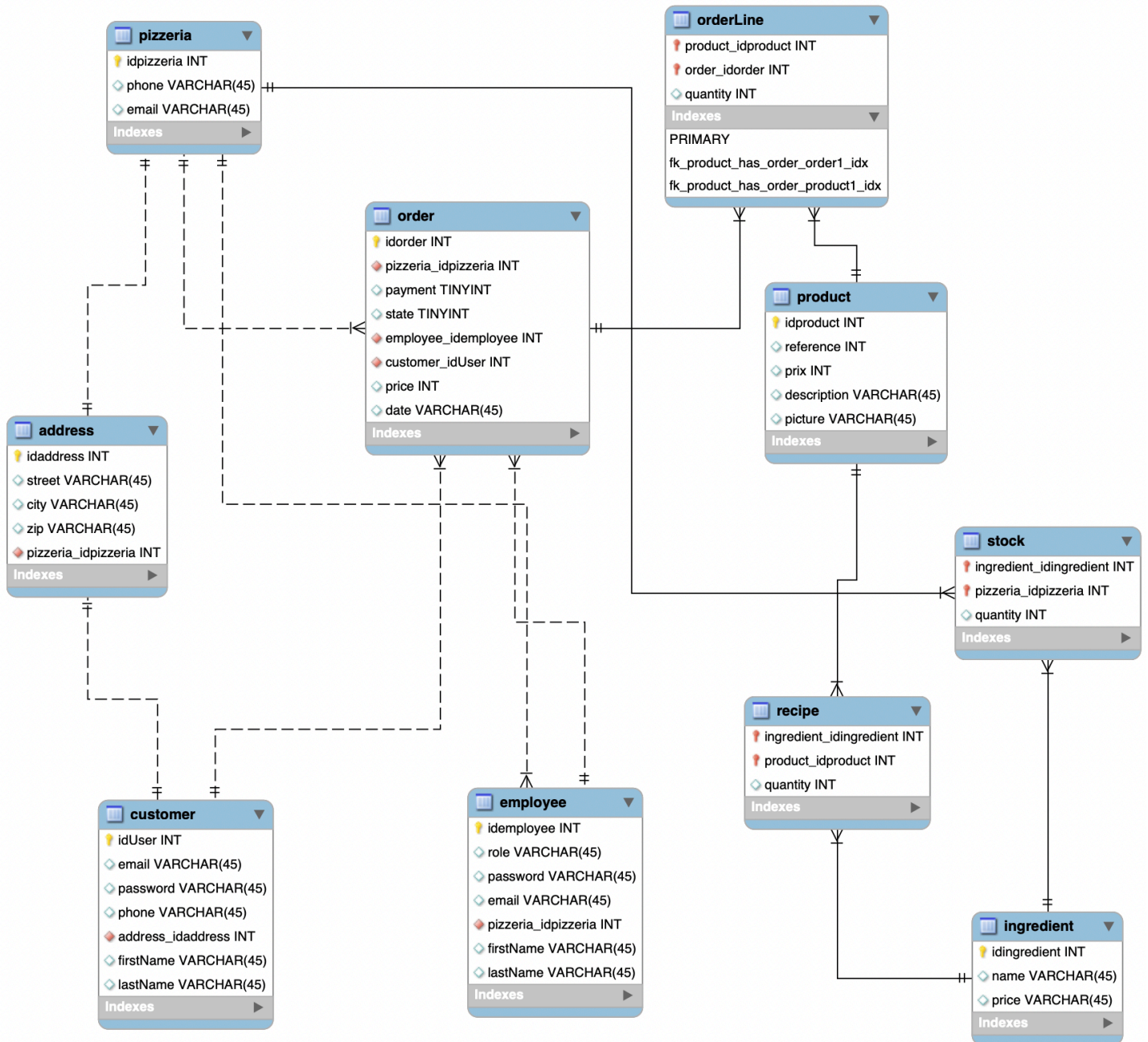
### 4.1.2. Modèle physique de données

Le modèle physique de données (MPD) permet de modéliser le domaine fonctionnel, à partir du diagramme de classes.

Les clés primaires, éléments uniques permettant d'identifier une donnée, et les clés étrangères, mettant en relation plusieurs tables entre elles, apparaissent sur ce diagramme.

Ce modèle va permettre de créer la structure finale de la base de données avec ses différents éléments.





## 5.ARCHITECTURE LOGICIELLE

### 5.1.Principes généraux

Les sources et versions du projet sont gérées par **Git**, les dépendances et le packaging par **symfony**.

#### *5.1.1.Les couches*

L'architecture applicative est la suivante :

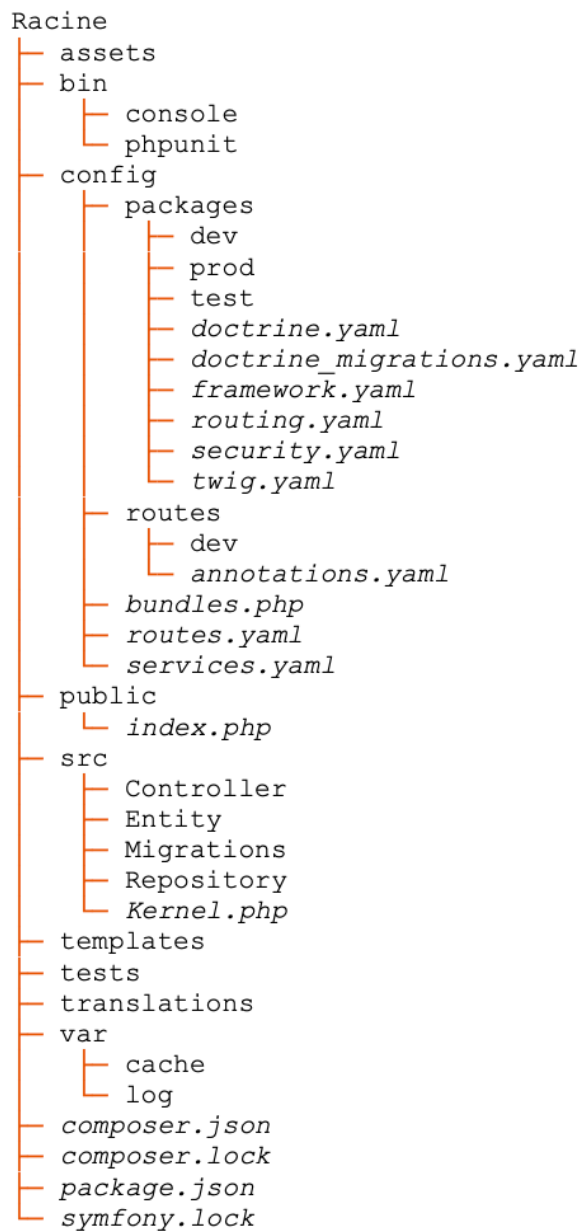
- **une couche View** : responsable de l'interface client/application ;
- **une couche Model** : responsable de la logique ;
- **une couche Controller** : responsable de la réception et de l'interprétation des interactions avec le client et de la réponse à lui donner via la vue au regard de la logique comprise dans le modèle.

#### *5.1.2.Les modules*

Les modules utilisées sont ceux du framework symfony, à savoir le module Doctrine qui est le module par défaut permettant de communiquer avec la base de données.

#### *5.1.3.Structure des sources*

La structuration des répertoires du projet suit la logique par défaut du framework symfony :



- **assets** : Ce dossier contient les différentes ressources graphiques dont le site web a besoin.
- **bin** : Ce dossier contient les exécutables dans le projet que ce soit ceux fournis avec la Framework (la console Symfony) ou ceux des dépendances (phpunit, simple-phpunit, php-cs-fixer, phpstan)
- **config** : Ce dossier contient toutes les configurations du site.
- **public** : Ce dossier contient les fichiers destinés à recevoir les requêtes utilisateurs.
- **src** : Ce dossier contient l'application, les controllers et différentes ressources nécessaires



à l'exécution du site.

- **templates** : Ce dossier contient les templates utilisés dans le site.
- **tests** : Ce dossier contient les différents tests du site.
- **var** : Ce dossier contient les fichiers de cache et de log.
- **vendor** : Ce dossier contient les dépendances du site et leurs loaders.
- **translations** : Ce dossier contient les traductions afin de rendre le site multi-lingual.

## 6.POINTS PARTICULIERS

### 6.1.Gestion des logs

Les logs permettent d'assurer un suivi dans l'utilisation du système. Ils permettent le suivi des évènements du site: erreurs, cause de crash...

Pour ce suivi, la bibliothèque Monolog sera utilisée. Cette bibliothèque permet de journaliser différentes informations lors de l'exécution de scripts php et de supprimer automatiquement les plus vieux.

Son intégration via symfony se fait de manière quasi automatique.

### 6.2.Fichiers de configuration

Les fichiers de configuration sont dans le dossier config.

La configuration du comportement du système se fait dans le fichier services.yaml.

### 6.3.Environnement de développement

L'environnement de développement est le suivant :

- **système d'exploitation** : MacOS (v. 11.2.2)
- **serveur local** :
  - Apache (v. 2.4.46);
  - MySQL (v. 8.0.23);
  - PHP (v. 8.0.2), symfony (v. 5.2.4) ;
- **création du MPD** : MySQL Workbench (v. 8.0.23) ;
- **IDE** : Sublime Text (v. 3.2.2).

## 6.4.Procédure de packaging / livraison

Le système fera l'objet d'un déploiement sur l'hébergeur OVH. Ce déploiement est la charge du client.

Les fichiers du système seront livrés avec les dumps de la fausse base de données.

Ils seront également stockés sur GitHub via un lien présent sur le PV de livraison.