

## **OC PIZZA**

# **Dossier de conception technique du Système de gestion de pizzerias**

Dossier de conception technique

Version 1.0

**KHADIRI YAZAMI Mehdi**  
Analyste Programmeur

## TABLE DES MATIERES

<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
3.1.1 - Package Client (sous système site web e-commerce) .....	6
3.1.2 - Package Pizzeria (sous-système — gestion pizzeria) .....	6
3.1.3 - Package Groupe OC Pizza (sous-système — gestion des pizzerias).....	6
3.1.4 - Package Authentification (sous-système — authentification) .....	6
<b>4 - Architecture de Déploiement.....</b>	<b>7</b>
<b>5 - Architecture logicielle.....</b>	<b>8</b>
5.1 - Principes généraux .....	8
5.1.1 - Structure des sources .....	8
<b>6 - Points particuliers .....</b>	<b>9</b>
6.1 - Gestion des logs .....	9
6.2 - Fichiers de configuration.....	9
6.3 - Ressources .....	9
6.4 - Environnement de développement .....	9
6.5 - Procédure de livraison .....	9

# 1 - VERSIONS

Auteur	Date	Description	Version
KHADIRI YAZAMI MEHDI	15/02/2018	Rédaction de conception technique	1.0

## 2 - INTRODUCTION

### 2.1 - Objet du document

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

L'objectif du document est d'informer et aider les développeurs pour la conception, la maintenance et l'évolution de l'application.

Les éléments du présent dossier découlent :

- Document du domaine fonctionnel
- Document des spécifications techniques
- Document du modèle physique de donnée

### 2.2 - Références

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

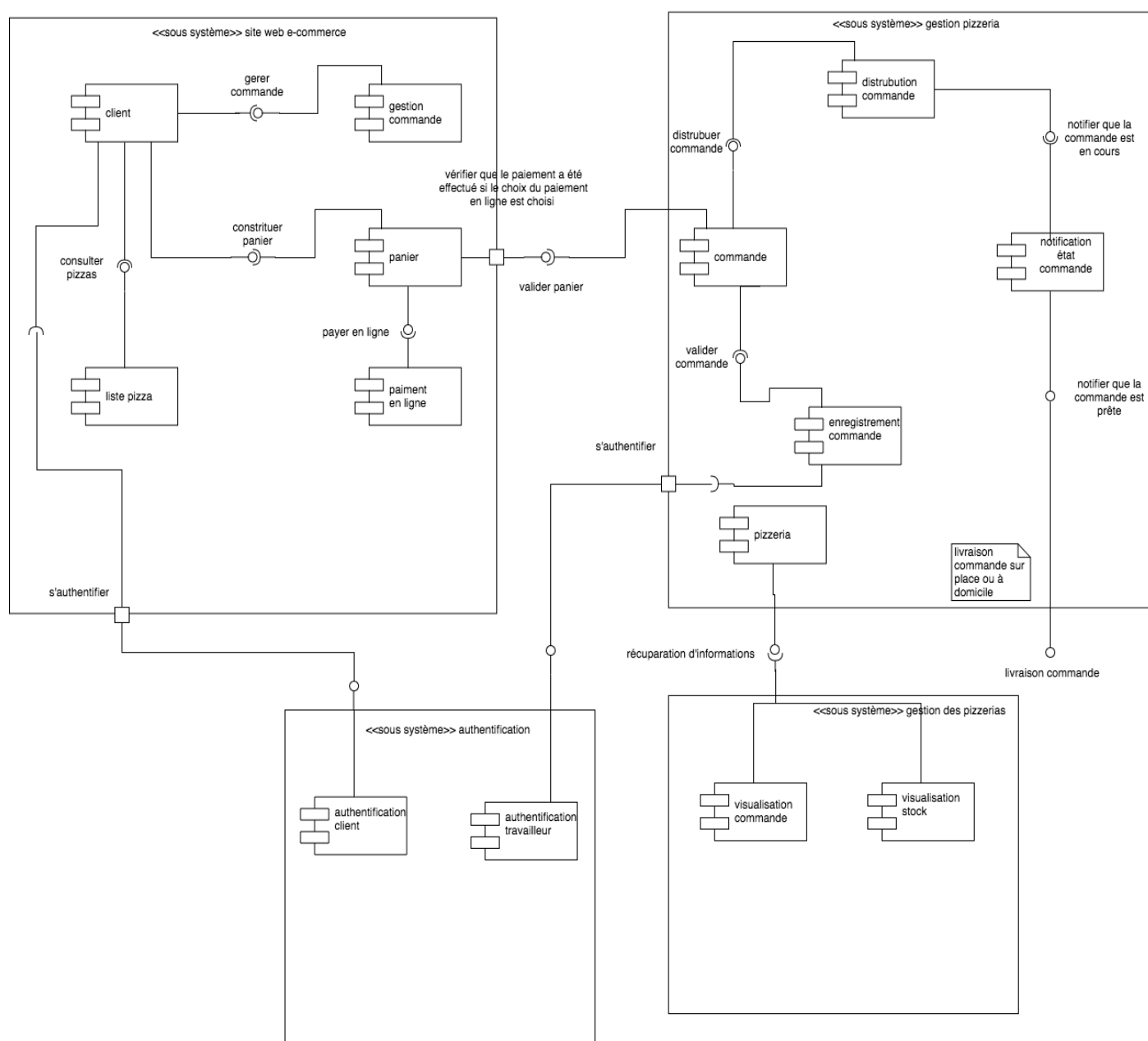
1. Dossier de conception fonctionnelle de l'application
2. Document des spécifications techniques
3. Document du modèle physique de donnée

## 3 - ARCHITECTURE TECHNIQUE

### 3.1 - Application Web

La pile logicielle est la suivante :

- **PHP 7.2.4** (Framework **Symfony 4**)
- SGBDR **MySQL 5.7.21**
- Serveur d'application HTTP **Apache 2.4**
- OS serveur **Ubuntu** : version 16.04 (64 bits)



### ***3.1.1 - Package Client (sous système site web e-commerce)***

#### ***3.1.1.1 - Composant client***

Permet au client de la pizzeria de s'authentifier, consulter la liste des pizzas, constituer un panier, payer en ligne et gérer les commandes via les différentes interfaces reliant les autres composants.

**Remarque** : Pour que le client puisse constituer un panier, commander, gérer les commandes, il doit obligatoirement être authentifié.

### ***3.1.2 - Package Pizzeria (sous-système — gestion pizzeria)***

#### ***3.1.2.1 - Composant commande***

Est le point d'entrée des commandes passées via le site web ou surplace, pour ensuite distribuer et gérer ces dernières.

#### ***3.1.2.2 - Composant notification état commande***

Permet aux différents employés de pizzerias de notifier où en est la commande.

#### ***3.1.2.3 - Composant enregistrement commande***

Permet à l'employé de pizzeria d'enregistrer une commande sur place, pour ce faire il doit s'identifier.

#### ***3.1.2.4 - Composant pizzeria***

Constitue les éléments d'une pizzeria comme le nom, l'adresse, l'identifiant...

### ***3.1.3 - Package Groupe OC Pizza (sous-système — gestion des pizzerias)***

#### ***3.1.3.1 - Composant visualisation commande & visualisation stock***

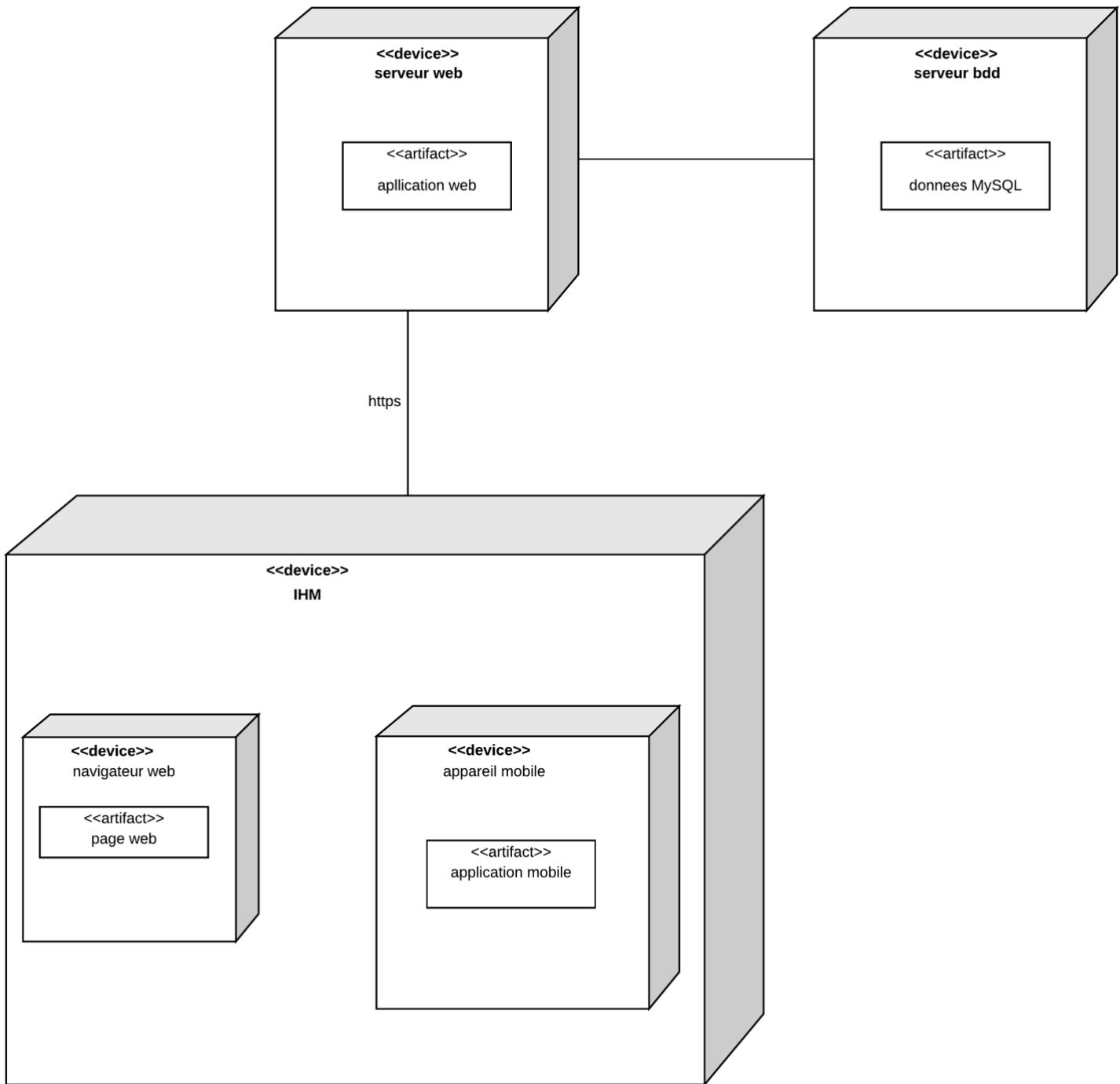
Ces deux composants ont pour but de visualiser les commandes enregistrées, en cours ainsi que les stocks d'une pizzeria sélectionnée au préalable.

### ***3.1.4 - Package Authentification (sous-système — authentification)***

#### ***3.1.4.1 - Composant authentification client & authentification travailleur***

Ces composants permettent soit au client ou au travailleur d'accéder aux composants nécessitant d'être authentifié.

# 4 - ARCHITECTURE DE DEPLOIEMENT



## 5 - ARCHITECTURE LOGICIELLE

### 5.1 - Principes généraux

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

#### 5.1.1 - Structure des sources

La structuration des répertoires du projet suit la logique suivante :

- Les répertoires sont créés de façon à respecter le concept du Framework Symfony :

```
/
├── app
├── bin
├── scr
├── tests
├── var
├── vendor
├── .gitignore
├── composer.json
├── composer.lock
└── web
```

- **app** : contient les configurations, template...
- **bin** : contient les scripts exécutables (commandes PHP)
- **scr** : contient les codes sources de l'application
- **tests** : contient les tests unitaires et fonctionnels
- **var** : contient les logs, les caches et les sessions générés par l'application
- **Vendor** : contient toutes les bibliothèques externes à l'application
- **web** : contient tous les fichiers CSS, JavaScript, images...



## 6 - POINTS PARTICULIERS

### 6.1 - Gestion des logs

Symfony organise les logs dans des canaux. Par défaut, il existe plusieurs canaux, incluant doctrine, les événements, la sécurité... Le canal est imprimé dans le log.

Symfony enregistre chaque message dans un seul fichier quel que soit le canal.

### 6.2 - Fichiers de configuration

La plupart des réglages de Symfony peuvent être configurés via un fichier de configuration écrit en YAML, ou avec du simple PHP.

#### 6.2.1.1 - Fichier *settings.yml*

settings.yml est le principal fichier de configuration.

Le fichier de configuration settings.yml pour une application peut être trouvé dans le répertoire app/APP\_NAME/config/.

*Pour les fichiers de configurations secondaires, se référer à la documentation officielle de Symfony.*

### 6.3 - Ressources

#### 6.3.1.1 - API Google Maps

Pour intégrer l'API Google Maps, se référer à la documentation officielle au lien suivant : <https://developers.google.com/maps/documentation/javascript/tutorial?hl=fr>

#### 6.3.1.2 - Images

Le client a fourni 4 images pour pouvoir livrer l'application opérationnelle.

#### 6.3.1.3 - Base de données

Deux fichiers SQL sont fournis avec les livrables, l'un contient les scripts SQL pour la création de la base de données et les différentes tables sans données, le deuxième contient le dump avec les données pour que le client puisse démarrer l'application de manière opérationnelle.

### 6.4 - Environnement de développement

L'IDE pour le projet Symfony est **PhpStorm**.

### 6.5 - Procédure de livraison

Concernant la procédure de livraison, se référer au document de *Description de déploiement des applications* dans les spécifications techniques.

Les identifiants et mots de passe des applications seront donnés au client après la mise en ligne et le paiement de ce dernier.