

**Client : OC Pizza**

**Projet : OC Pizza**

Dossier de conception technique

Version 1

**Auteur**

Guillaume Toussaint

*Développeur*

## 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 - Présentation de la solution .....	5
3.2 - Model physique de données .....	6
3.2.1 - <i>Diagramme</i> .....	6
3.2.2 - <i>Explication</i> .....	7
3.2.3 - <i>Script de création</i> .....	7
3.2.4 - <i>Jeu de données</i> .....	7
3.2.5 - <i>Requêtes</i> .....	7
3.3 - Diagramme de composants .....	8
3.3.1 - <i>Diagramme</i> .....	8
3.3.2 - <i>Explication du diagramme</i> .....	8
3.3.2.1 - Subsystem Front-end.....	8
3.3.2.2 - Subsystem Back-end .....	9
3.3.2.3 - Subsystem Database .....	9
3.4 - Choix technique .....	10
3.4.1 - <i>Front-end : Application web</i> .....	10
3.4.1.1 - Url .....	10
3.4.1.2 - Considérations techniques .....	10
3.4.2 - <i>Back-end : API REST</i> .....	11
3.4.2.1 - Communication Front-end <-> Back-end .....	12
3.4.2.2 - Communication Back-end <-> Bdd.....	12
3.4.3 - <i>Base de données</i> .....	12
3.4.3.1 - PostgreSQL.....	12
3.5 - Diagramme de déploiement.....	13
3.5.1 - <i>Diagramme</i> .....	13
3.5.2 - <i>Explication du diagramme</i> .....	14
3.6 - Hébergement .....	14
<b>4 - Glossaire .....</b>	<b>15</b>

# 1 - VERSIONS

Auteur	Date	Description	Version
Guillaume Toussaint	16/08/2021	Création du document	1

## 2 - INTRODUCTION

### 2.1 - Objet du document

Le présent document constitue le dossier de conception technique de l'application OC Pizza. Il présente la solution technique proposé pour réaliser le projet OCPizza. Les éléments du présent dossier découlent :

- De l'analyse des besoins client
- De la conception fonctionnelle réalisé préalablement
- De l'expertise et de l'expérience de IT Consulting & Développement dans le domaine du R&D logiciels.

### 2.2 - Références

Se référer également aux éléments suivants, les diagrammes sont disponibles en haute résolution :

1. **Diagramme de classes,**
2. **Document dossier de conception fonctionnelle, (cf. §4 Diagramme de classes et explication du diagramme de classes)**
3. **Diagramme Modèle physique de données,**
4. **Document de description du modèle physique de données et du jeu de données,**
5. **Diagramme de composants,**
6. **Diagramme de déploiement,**
7. **Script de création de la structure de la base de données,**
8. **Script de création du jeu de données,**
9. **Script exemples de requêtes SQL**

## 3 - ARCHITECTURE TECHNIQUE

### 3.1 - Présentation de la solution

La solution proposée est une solution « **full web** ».

Il s'agira d'un site Web Responsive Design accessible depuis les principaux terminaux actuellement utilisés : ordinateur, tablette, téléphone portable.

Le site sera donc compatible avec les différents systèmes d'exploitation du marché (Windows, Apple, Linux, Android). On y accédera avec n'importe quel navigateur « moderne » (Chrome, Safari, Firefox, Microsoft Edge).

La solution sera composée de :

- Une **interface Web** dédiée aux clients et aux employés
- D'une partie **serveur fonctionnant avec une API REST**, la sécurisation des utilisateurs, les process métier et les accès base de données.
- D'une **base de données postgresql** unique qui centralise toutes les informations du groupe

L'ensemble de l'application sera développé par IT C&D en utilisant des Framework de l'industrie.

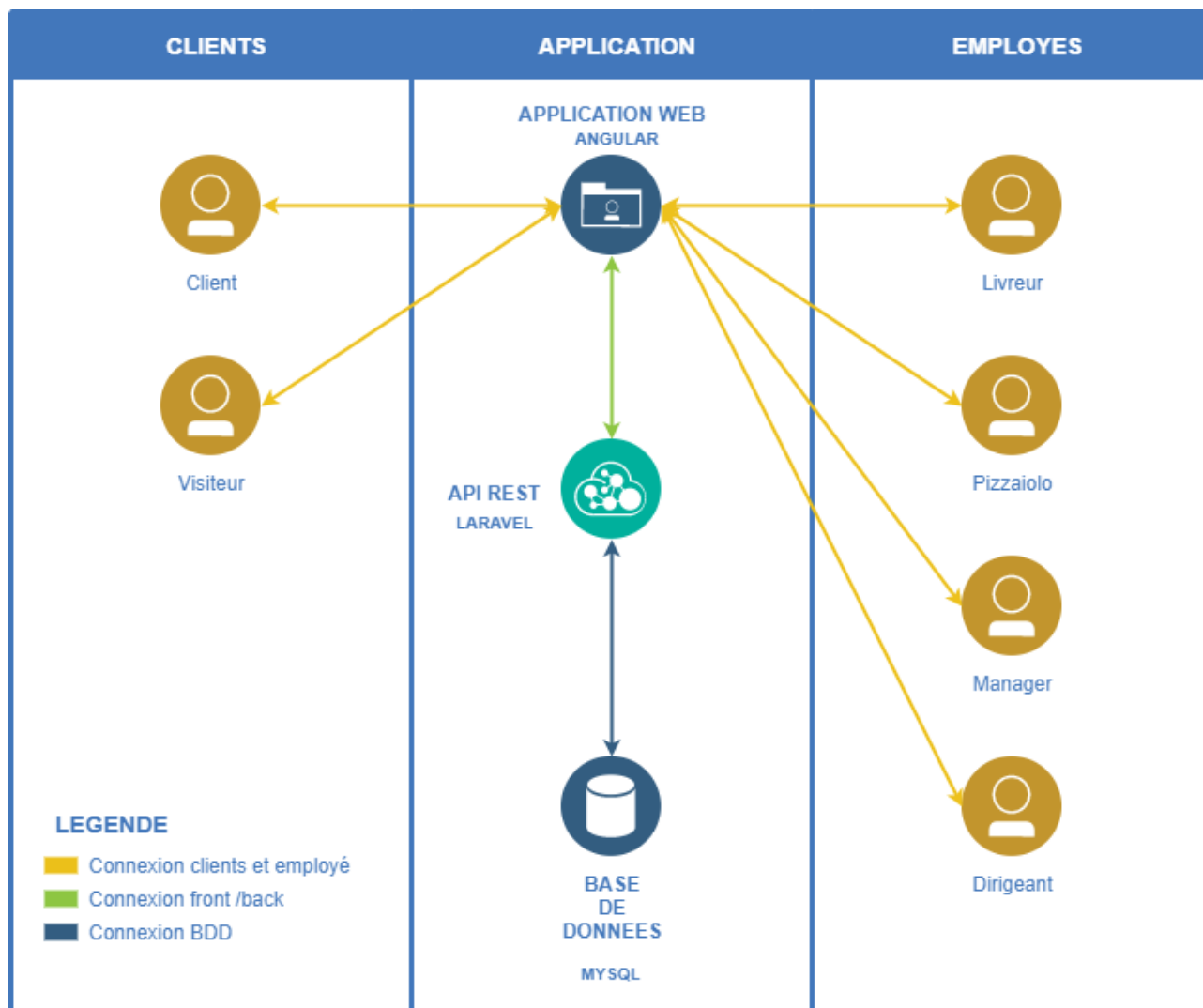


Figure 1 Organisation de la solution

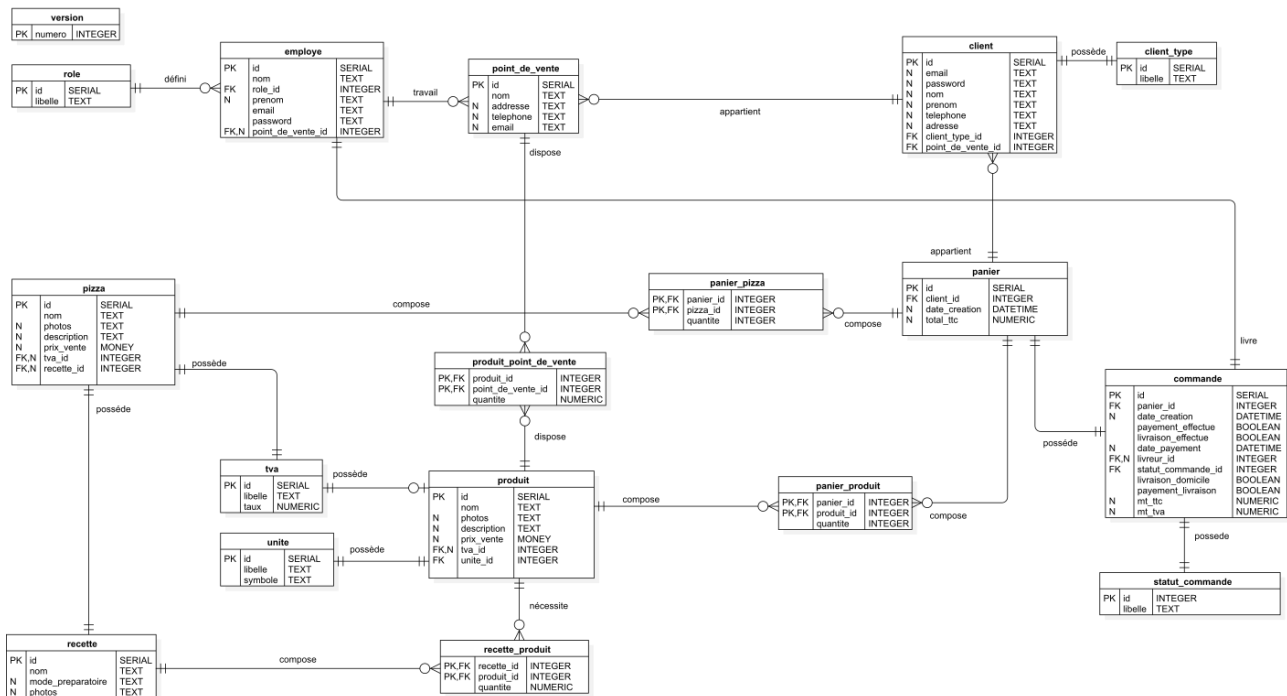
## 3.2 - Model physique de données

### 3.2.1 - Diagramme

Le diagramme du modèle physique de données est disponible en grande résolution dans le fichier « model\_physique\_de\_donnees.pdf ».

Ci-dessus un aperçu du fichier.

Oc Pizza - Modèle physique de données



### 3.2.2 - Explication

Une explication détaillée des tables, de leur rôle et de leur relation est disponible dans le fichier annexe « description\_du\_modele\_physique\_de\_donnees.pdf ».

### 3.2.3 - Script de création

Le script de création de la base de données est disponible dans le fichier « mydata\_table\_create.sql ».

### 3.2.4 - Jeu de données

Le script d'insertion du jeu de données est disponible dans le fichier « insert.sql ».

### 3.2.5 - Requêtes

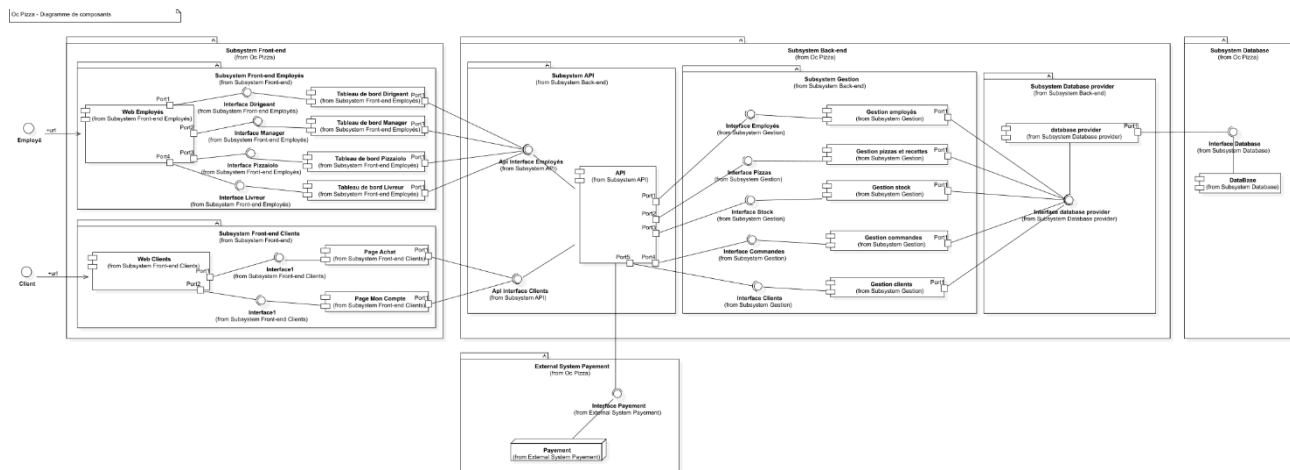
Des exemples de requêtes sont disponibles dans le fichier « select.sql ».

## 3.3 - Diagramme de composants

### 3.3.1 - Diagramme

Le diagramme de composant est disponible en grande résolution dans le fichier « diagramme\_de\_composants.pdf ».

Ci-dessous un aperçu du fichier.



### 3.3.2 - Explication du diagramme

Le diagramme de composant est découpé en quatre parties. Chaque partie correspondant à un élément majeur du système.

#### 3.3.2.1 - Subsystem Front-end

Ce module concerne la partie *interface utilisateur* du système. Elle expose deux interfaces, une pour les clients et une pour les employés. Chaque interface sera accessible par une url.

- **Subsystem Front-end Clients** : C'est le sous-système réservé aux clients. Ce sous-système se compose de deux nœuds :
  - Page « Achat » : Cette page permettra au client de parcourir les produits disponibles est de réaliser ses achats.
  - Page « Mon Compte » Cette page permettra au client de gérer son compte client.
- **Subsystem- Front-end Employés** : C'est le sous-système réservé aux employés et aux dirigeants. Ce sous-système se compose de quatre nœuds qui correspondes aux rôles des employés et des dirigeants. Chaque rôle est matérialisé par un tableau de bord :
  - Tableau de bord Dirigeant
  - Tableau de bord Manager



- Tableau de bord Pizzaiolo
- Tableau de bord Livreur

### 3.3.2.2 - Subsystem Back-end

Ce module concerne la partie « Métier » du système. Il est découpé en quatre sous-systèmes :

- **Subsystem API** : Ce sous-système expose des interfaces permettant la communication avec le front-end. Cette communication se fera à travers une API rest. Cette API sera divisée en deux parties.
  - Une interface employée
  - Une interface clients.
- **Subsystem Gestion** : Ce sous-système est responsable de la gestion métier. Il exerce cette responsabilité à travers des nœuds spécialisés dans les rôles principaux de l'application.
  - Gestion des employés
  - Gestion des pizzas et de leurs recettes
  - Gestion du stock
  - Gestion des commandes clients
  - Gestion des comptes clients
- **Subsystem Database provider** : Ce sous-système permet la communication avec la base de données.
- **External system Payement** : C'est la partie communication avec l'interface de payement. Il sera dépendant du choix du prestataire de payement et l'interface qu'il mettra à disposition.

### 3.3.2.3 - Subsystem Database

Ce sous système est la base de données.

## 3.4 - Choix technique

### 3.4.1 - Front-end : Application web

Il s'agit ici de la partie interface utilisateur de l'application. **C'est l'interface graphique** qui sera utilisé par les clients et les employés. Elle sera réalisée pour le web.

- Pour les clients c'est le site de vente en ligne
- Pour les employés ce site permettra de gérer les points de vente et le groupe,
  - **Commandes** et **livraisons** client
  - Gestion des **stocks**
  - Gestion des **recettes**
  - **Administration** des points de vente et des comptes employés

#### 3.4.1.1 - Url

- Les clients se connecteront sur l'url par défaut du site, par exemple <https://www.ocpizza.com>.
- Les employés se connecteront sur une url spécifique du site, Une route leur sera dédié. Par exemple [https://www.ocpizza.com/production/\[idemploye\]](https://www.ocpizza.com/production/[idemploye]). Cette url comportera une route « /production » qui dirigera l'employé vers la partie gestion. Cette url sera interdite aux clients. Un paramètre supplémentaire [idemploye] identifiera l'employé et personnalisera son interface en fonction de son rôle. Par exemple :
  - Un livreur verra son tableau de bord de livraison.
  - Un pizzaiolo ou un manager verra l'interface de gestion des commandes.
  - Un dirigeant verra le tableau de bord du groupe.

#### 3.4.1.2 - Considérations techniques

Techniquement cette interface sera réalisée en HTML/Javascript/CSS. Pour cela on s'appuiera sur le **Framework Angular**.

- Angular est Framework développé par Google qui permet la création d'application web.
- Angular permet de développer des **applications dynamiques** et responsives.
- Angular est inclus dans le gestionnaire de paquet de Node.js C'est une solution gratuite est facilement installable. **Il n'y a pas de cout de licence à prévoir.**
- Techniquement Angular utilise le langage **TypeScript**. La finalité de TypeScript est de générer le Javascript qui sera utilisé dans le navigateur. L'intérêt du TypeScript et de

proposer une approche plus moderne du Javascript ce qui apporte une plus grande fiabilité au programme.

- TypeScript type le langage rendant ainsi son écriture plus sécurisé et compréhensible.
- TypeScript utilise les fonctions lambda, ce qui permet des écritures plus concises.
- TypeScript permet la création de classe rendant le code plus modulaire.
- MVC
- **Hébergement Angular**
  - La mise en production d'un site Angular utilisant un serveur REST ne nécessite qu'un simple serveur web statique. Son coup de mise en production est limité.
  - Une solution est d'utiliser un serveur **Amazon S3**. Amazon S3 permet par un simple glissé/déposer de fichier de publier une solution Angular.

### 3.4.2 - Back-end : API REST

Le cœur de l'application sera basé sur le principe d'une API REST. Une API REST permet d'échanger des données entre un poste client et un serveur à travers une connexion internet. Les échanges se font en http d'une façon standard. Les données seront transportées en JSON.

Pour implémenter cette API REST nous allons utiliser **php** et le Framework **Laravel**.

Les avantages de **Laravel** sont :

- Solution stable et éprouvée depuis 2011
- Libre de droits
- Dispose d'une très bonne documentation
- Framework orienté objet et MVC, même approche programmatique que pour le front.
- Gestion des routes (url)
- Gestion des authentifications
- Sécurisation des routes
- Sécurisation des enregistrements des mots de passe
- Sécurisation des injection SQL
- Bonnes compatibilités avec PostgreSQL
- La connexion se paramètre simplement dans le fichier .env
- Possibilité d'utiliser directement le langage SQL

- Possibilité d'utiliser l'ORM Eloquent
- **Hébergement Laravel**
  - Laravel est basé sur php, son hébergement se fait sur un serveur web disposant d'un interpréteur php. Des solutions comme Apache ou Nginx mettent à disposition cet interpréteur.
  - Amazon propose des machines clés en main pour ce type d'hébergement, par exemple **Amazon EC2** avec php.

### 3.4.2.1 - Communication Front-end <-> Back-end

Le Back-end exposera une API REST sur une URL définie. Les échanges entre le Front-end et le Back-end se feront en HTTPS sur le port 443. Les données transmises seront formatées en JSON.

### 3.4.2.2 - Communication Back-end <-> Bdd

#### Connexion PostgreSQL avec Laravel :

Par défaut Laravel est configuré pour fonctionner avec MySQL, aussi il faudra modifier la configuration de Laravel pour le faire communiquer avec PostgreSQL. Pour cela :

- **Installer le driver** PostgreSQL pour PHP : php-pgsql.
- **Activer l'extension** pdo\_pgsql dans le fichier php.ini
- **Paramétrer le fichier .env** pour utiliser pgsql. Db\_CONNECTION=pgsql.

### 3.4.3 - Base de données

Un point essentiel de la solution est la centralisation des données du groupe dans une seule base de données. Nous allons utiliser une **base données relationnelle PostgreSQL**. PostgreSQL est un serveur de base de données éprouvé et très répandu qui offre toutes les fonctionnalités et les garanties nécessaires pour ce type d'utilisation.

#### 3.4.3.1 - PostgreSQL

Les avantages de **PostgreSQL** :

- Base de données relationnelle et objet
- Reconnu pour sa stabilité
- Multiplateforme, fonctionne sous Linux, Mac OS, et Windows
- Conformité avec le langage SQL
- Gratuit, Open source, Licence BSD (Berkeley Software Distribution Licence)

**Hébergement** : Amazon propose des services de base de données « clés en main » **Amazon**

## RDS.

**Port** : Le port par défaut de PostgreSQL est le port 5432.

**PgAdmin** : PgAdmin est une interface graphique permettant d'interagir avec la base de données.

**Psql** : « *psql est une interface en mode texte pour PostgreSQL. Il vous permet de saisir des requêtes de façon interactive, de les exécuter sur PostgreSQL et de voir les résultats de ces requêtes. Alternativement, les entrées peuvent être lues à partir d'un fichier ou à partir des arguments de la ligne de commande. De plus, il fournit un certain nombre de métacommandes et plusieurs fonctionnalités style shell pour faciliter l'écriture des scripts et automatiser une grande variété de tâches.* »

**Createdb** : createdb est un programme en ligne de commande permettant la création de base de données.

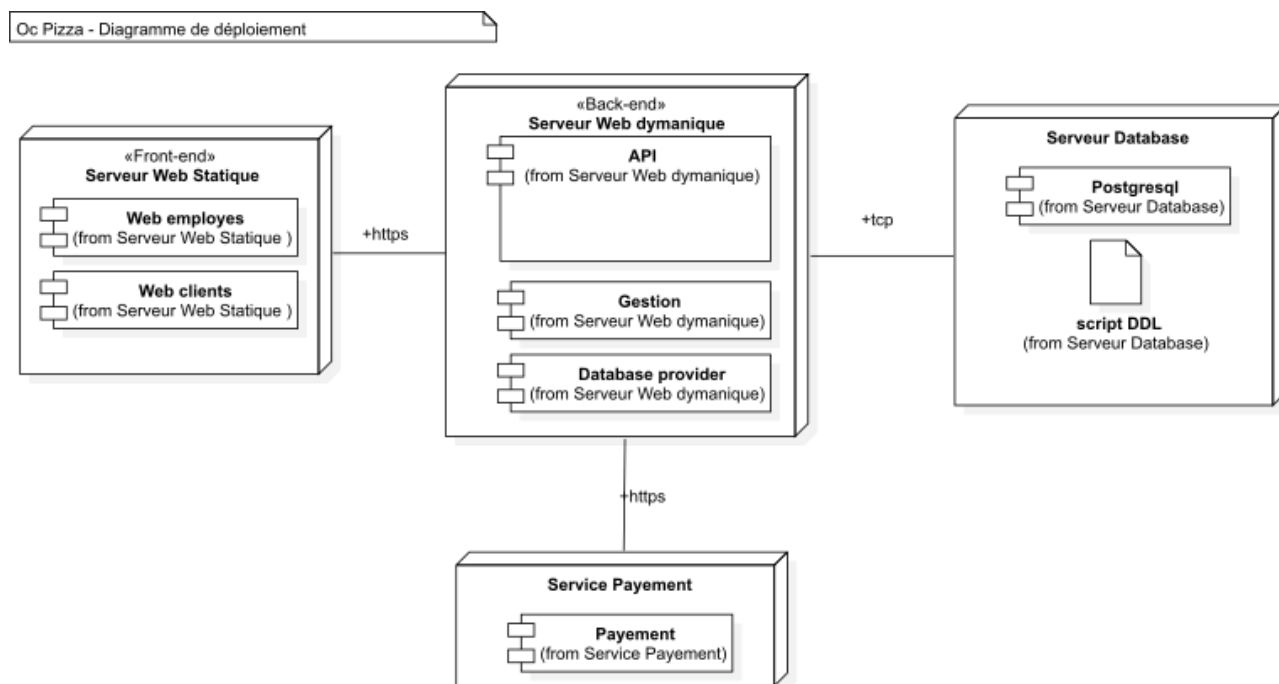
**Dropdb** : dropdb est un programme en ligne de commande permettant la suppression de base de données.

## 3.5 - Diagramme de déploiement

### 3.5.1 - Diagramme

Le diagramme de déploiement est disponible en grande résolution dans le fichier « diagramme\_de\_deploiement.pdf ».

Ci-dessus un aperçu du fichier.



### 3.5.2 - Explication du diagramme

- **Front-end** : Le front-end est constitué des fichiers HTML/CSS/Javascript hébergé sur un serveur web. De par le choix technique d'utiliser **Angular** ce sont des fichiers statiques. L'exécution du code se faisant sur les postes clients. Le front-end communiquera avec le back-end en https sur l'url du back-end.
- **Back-end** : Le back-end est constitué par un serveur web php exposant une API REST. Il contient le code métier et communique avec la base de données. Il gérera également les échanges avec le module externe de paiement. Il disposera d'une url permettant au front-end de communiquer avec lui. Les échanges avec la base de données seront exécutés par le driver de base de données sur la couche tcp.
- **Serveur Database** : la base de données sera hébergée sur un serveur de base données. Le port de communication (5432) devra être ouvert.
- **Service Paiement** : C'est le service du prestataire de paiement par internet. Par exemple Paypal.

## 3.6 - Hébergement

L'ensemble de l'application peut être hébergé chez Amazon avec le découpage suivant :

- **Front : Amazon S3**. S3 pour Simple Storage Service, c'est un service de stockage d'objet sur internet, ce service est adapté à l'hébergement d'un site web statique comme une application Angular. Il est d'une grande fiabilité et disponibilité.
- **Back : Amazon EC2**, Environnement d'exécution préconfiguré par Amazon, Amazon Machine Images (AMI) intégrant une distribution linux, un serveur web et un interpréteur php. Ces machines disposent en outre de composants d'infrastructure générique nécessaires à la sécurité d'un tel environnement, connexion sécurisés (ssh, scp), pare-feu, réseaux virtuels, montée en charge. Les capacités physiques sont ajustables en fonction de la charge de travail. L'environnement pourra s'adapter à l'évolution de l'entreprise.
- **Base de données : Amazon RDS**. Relational Database Service (RDS) est un service de mise à disposition de base de données. Il permet l'accès à plusieurs types de moteur de base données dont **PostgreSQL**.

## 4 - GLOSSAIRE

<b>Full Web</b>	Conception qui permet d'avoir accès à un logiciel grâce à une simple connexion à internet via n'importe quel navigateur web.
<b>Responsive Design</b>	Le Responsive Design est une technique de conception d'interface digitale qui fait en sorte que l'affichage d'une quelconque page d'un site s'adapte de façon automatique à la taille de l'écran du terminal qui le lit.
<b>Front-end</b>	Partie frontale d'une application web. Conception du design de l'interface graphique.
<b>Back-end</b>	Partie serveur d'une application web. Conception des aspects serveur et métier, gestion de la base de données
<b>API REST</b>	Application web fonctionnant sur un principe de demande et de délivrance de données. Par exemple un poste client peut demander à un serveur la liste des employés, le serveur répond en renvoyant la liste. La réponse est uniquement constituée des données. Il n'y a pas de notion de mise en page, c'est ensuite au poste client d'afficher la liste. Ce type de fonctionnement permet de dissocier le traitement des données de leur affichage.
<b>Node.js</b>	Environnement d'exécution axé vers javascript. Node.js est nécessaire pour la mise en œuvre d' Angular.
<b>Angular</b>	Angular est une Framework open source développée par Google, Il permet la création d'application web. Il travaille sur la partie cliente d'une application.
<b>PHP</b>	HypertextPreprocessor. Langage de programmation interprété et open source spécialisée dans le développement d'application web. Un serveur web désirant générer des pages web dynamique peut utiliser php. Php est un des langages les plus utilisés au monde pour générer des site web. Php est nécessaire à l'exécution de Laravel
<b>Laravel</b>	Laravel est un Framework open source basée sur php. Il permet la création d'application web.
<b>SQL</b>	Structured Query Language. Langage d'exploitation de base de données. Il permet la gestion de la structure de données et la gestion des données en elle-même.
<b>ORM</b>	Mapping Objet-Relationnel. Élément de code informatique destiner a facilité la manipulation des données. Il se place entre le programme et la base de données.



IT C&D

**IT Consulting & Development**

S.A.R.L. au capital de 1 000,00 € enregistrée au RCS de Xxxx – SIREN 999 999 999 – Code APE : 6202A