

Vous



Nous



# Dossier de conception technique

Version 1

**Auteur**  
**Frédéric Pichot**  
**Concepteur Développeur de Solution Digitale**



# Table des MATIÈRES

<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 - Composants généraux	6
3.1.1 Diagramme de classes	6
3.1.2. Détail des relations	6
3.1.3 Diagramme de composants	9
3.1.4 Description des composants	9
3.2 - Application Web	11
3.2.1 Composants HTML 5	11
3.2.2 Composants CSS3	11
3.2.3 JAVASCRIPT	11
3.2.3 Sublime Texte	11
3.3 - Applications nomades	12
3.3.1 Android Studio	12
3.3.2 Kotlin sous Android Studio	12
3.3.3 Swift sous Xcode langage	12
3.3.4 Swift sous Xcode langage	12
<b>4 - Architecture de Déploiement</b>	<b>13</b>
4.1 - Diagramme général de déploiement	13
4.2 - Détail du diagramme	14
4.2.1 - Serveur de Base de donnée	14
4.2.2 - L'API REST	14
4.2.3 Le serveur NGINX	14
4.2.4 Le Système de base de données	15
4.2.5 PHP le logiciel d'interface avec MySQL	15
4.2.5 Apache	15
<b>5 - Architecture Logicielle</b>	<b>16</b>
5.1 - Principe généraux le MPD	16
5.2 - Détail du MLD	16
5.2.1 Les tables commande, pizza, facture et ligne de commande	16
5.2.2 Les tables pizza, recette, stock et ingrédient	16



5.2.3 Les tables commande, statut commande, retrait, paiement	17
5.2.4 Les tables pizzeria, formule et pizza	17
5.2.5 Les tables commande, pizza, facture et ligne de commande	17
5.2.6 Les tables commande, pizza, facture et ligne de commande	18
<b>6 - Points particuliers</b>	<b>18</b>
6.1 - Gestion des logs	18
6.2 Fichiers de configuration	18
6.2.1 Application web	18
6.2.2 - Datasources	19
6.3 - Ressources	19
6.4 - Environnement de développement	19
6.5 - Procédure de packaging / livraison	19
6.5.1 Livraison	19
6.5.2 Formation	19
6.5.3 Suivi	19
6.5.4 Problème d'implémentation	20
6.5.5 Mise à jour	20



# 1 - Versions

Auteur	Date	Description	Version
Frédéric Pichot	21/10/2019	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 ».

L'objectif de ce document est d'expliquer les choix techniques qui ont été sélectionnés pour permettre à l'entreprise OC-Pizza de couvrir la totalité des besoins actuels tout en permettant une évolution future.

Les éléments du présents dossiers découlent :

- du rendez-vous du mois de juin et des échanges téléphoniques
- de la rencontre de juillet confortée par les correspondances numériques

### 2.2 - Références

Pour de plus d'information, le lecteur est invité à se référer aux éléments suivants :

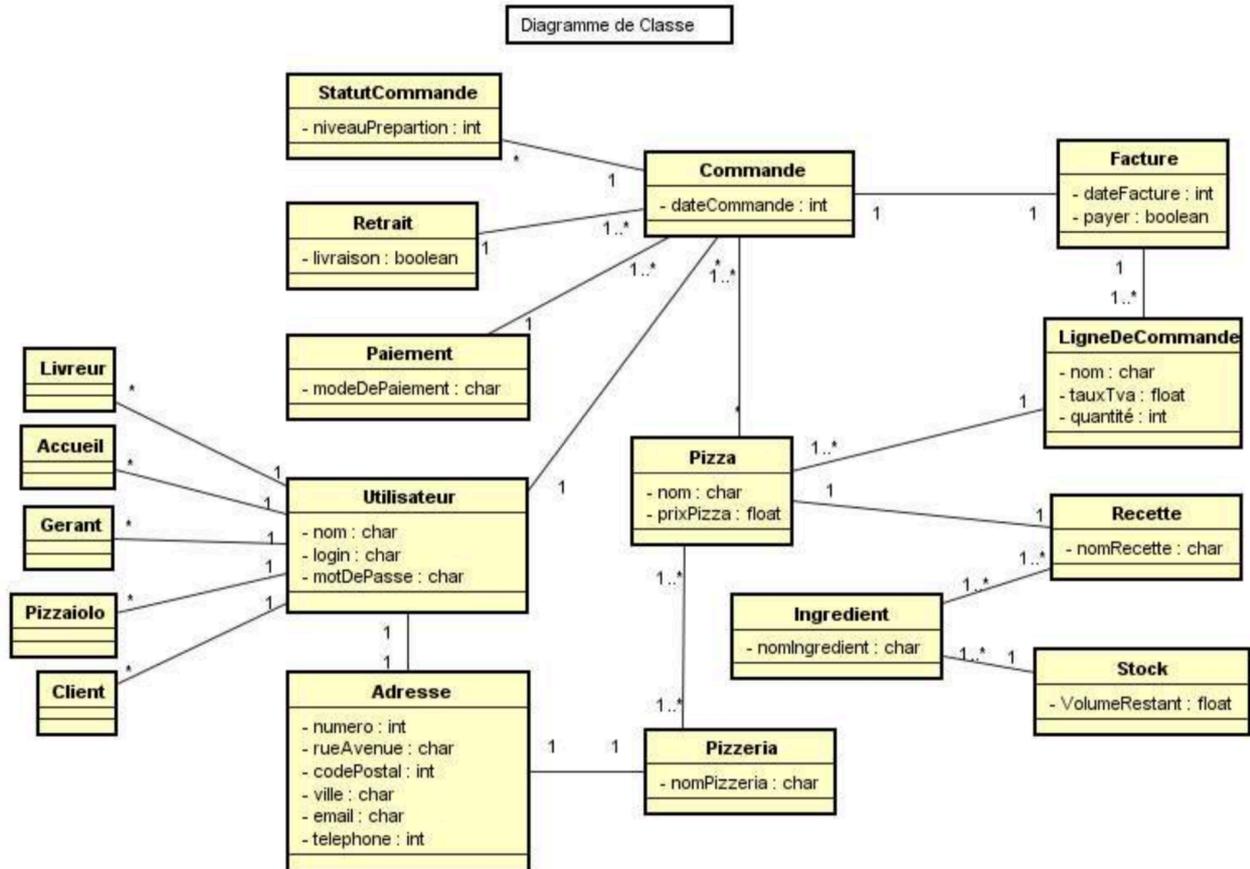
1. DCT - Dossier de Conception Technique
2. DCF - Dossier de Conception fonctionnelle



# 3 - Architecture Technique

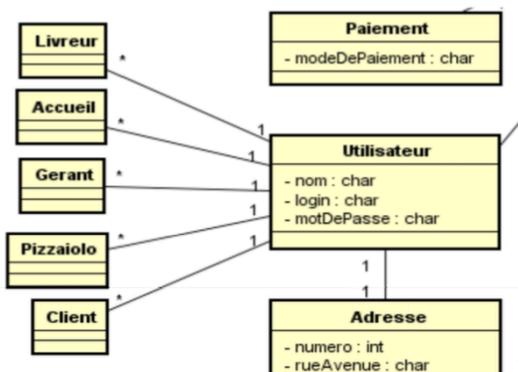
## 3.1 - Composants généraux

### 3.1.1 Diagramme de classes



### 3.1.2. Détail des relations

#### 3.1.2.1 Utilisateur, Livreur, Accueil, Gérant, Pizzaiolo et Client

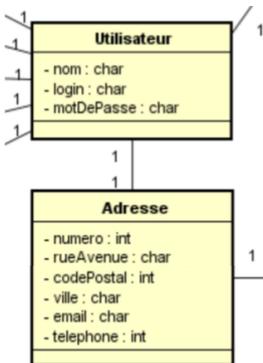


Pour permettre une modélisation, chacun des protagonistes cités sont potentiellement un utilisateur du système.

La relation qui les unit, à savoir \*/1, stipule que chaque livreur ou autres ne peut-être qu'un seul utilisateur. Au moment de l'action, mais l'objet utilisateur peut avoir plusieurs objets du même type



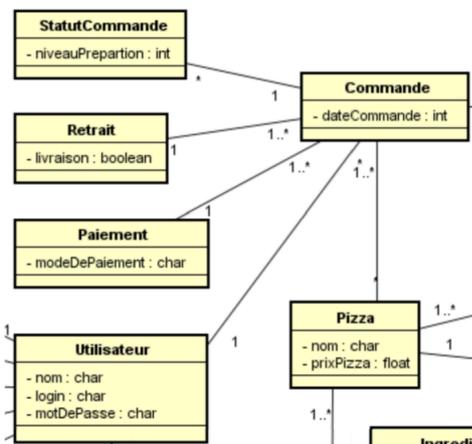
### 3.1.2.2 Utilisateur et Adresse



Cette partie du graphe mentionne que l'utilisateur est décrit par les attributs : nom, login et motDePasse. En effet, l'utilisateur pour pouvoir consommer devra s'enregistrer puis s'identifier à chaque fois qu'il se connecte. Les coordonnées de l'utilisateur s'obtiendront par le lien qui relie la table « Utilisateur à celle d' « Adresse.

Cette liaison est du type 1/1, chaque utilisateur n'a qu'une seule adresse et réciproquement une adresse n'appartient qu'à un seul utilisateur

### 3.1.2.3 Commande, Statut Commande, Retrait, Paiement, Utilisateur et Pizza



Cette dernière partie regroupe les classes Commande, Statut, Retrait, Paiement, Utilisateur et Pizza. La liaison entre les deux dernières est du type \*/1 vu que la commande peut avoir plusieurs statuts, mais pour que chaque pizza le statut est unique.

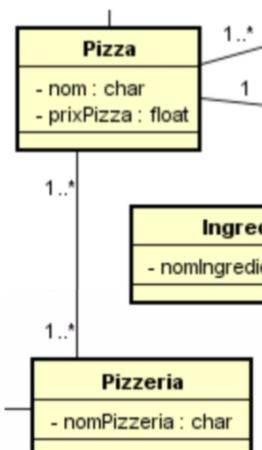
L'attribut niveau de préparation est imagé par un numéro qui pourrait aller de 1 à 5, dont le 1 serait la prise de commande et le 5 le paiement.

La classe Retrait, un seul attribut indique si la commande est à livrer ou non, mais pour une même commande il n'y en aura qu'un.

La Commande ne peut avoir qu'un seul utilisateur, mais celui-ci peut passer plusieurs commande.

Le mode de paiement est un « char » pour permettre de régler en cash, en chèque ou en carte voir d'autres qui seront à définir

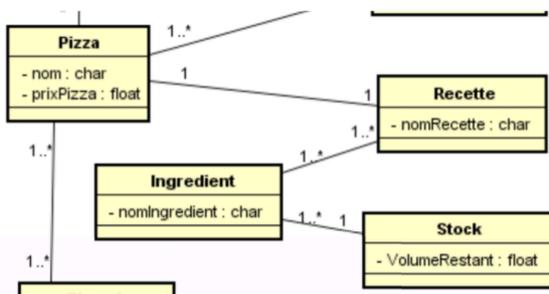
### 3.1.2.4 Pizzeria et Pizza



La relation qui unit Pizzeria et pizza est du type 1/\* car chaque pizzeria a plusieurs modèles de pizza à proposer. Comme d'ailleurs chaque pizza est fabriquée dans toutes les pizzerias du groupe. Ici le « \* » aurait pu être remplacé par le chiffre cinq au regard du nombre d'établissement que dispose le groupe actuellement.



### 3.1.2.5 Pizza, Recette et Stock

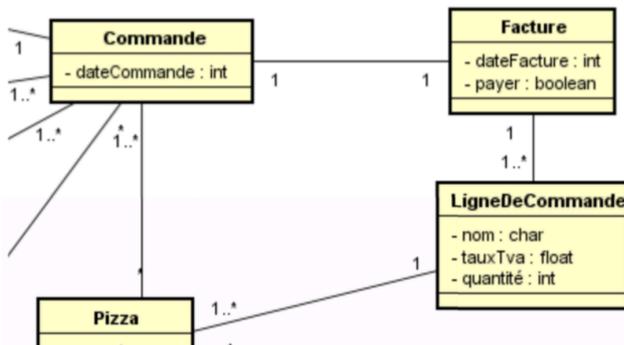


La pizza à un nom comme attribut, mais sa recette est dans une autre table car la composition regroupe plusieurs ingrédients. Pour autant, leur liaison est du type 1/1 car une pizza n'a qu'une seule recette et celle-ci définit une unique pizza.

Celle de Recette et Ingrédient est du type 1/\* dans les deux sens puisque chaque ingrédient peut être présent dans différentes recettes et en retour chaque recette contient plusieurs ingrédients.

Le stock quant à lui à plusieurs ingrédients qui a lui qu'un seul niveau de stock.

### 3.1.2.6 Commande, Facture et Ligne De Commande

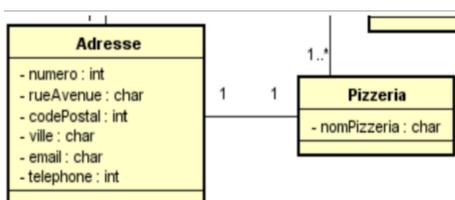


Cette partie regroupe la commande, la facture, la ligne de commande et la pizza. La liaison qui unit pizza et ligne de commande précise que sur une même ligne de commande il est possible d'avoir une ou plusieurs pizzas, mais le nom de la pizza ne peut-être inscrit qu'une seule fois.

La liaison facture ligne de commande suit les mêmes règles que précédemment, une ligne de commande ne peut se trouver qu'une seule fois par contre la facture peut avoir plusieurs lignes de commande. De plus, une facture ne peut être attachée qu'à une seule commande et celle-ci ne peut faire éditer qu'une facture.

L'attribut booléen payé n'offre que deux statuts soit elle est réglée plu non.

### 3.1.2.3 Pizzeria et Adresse

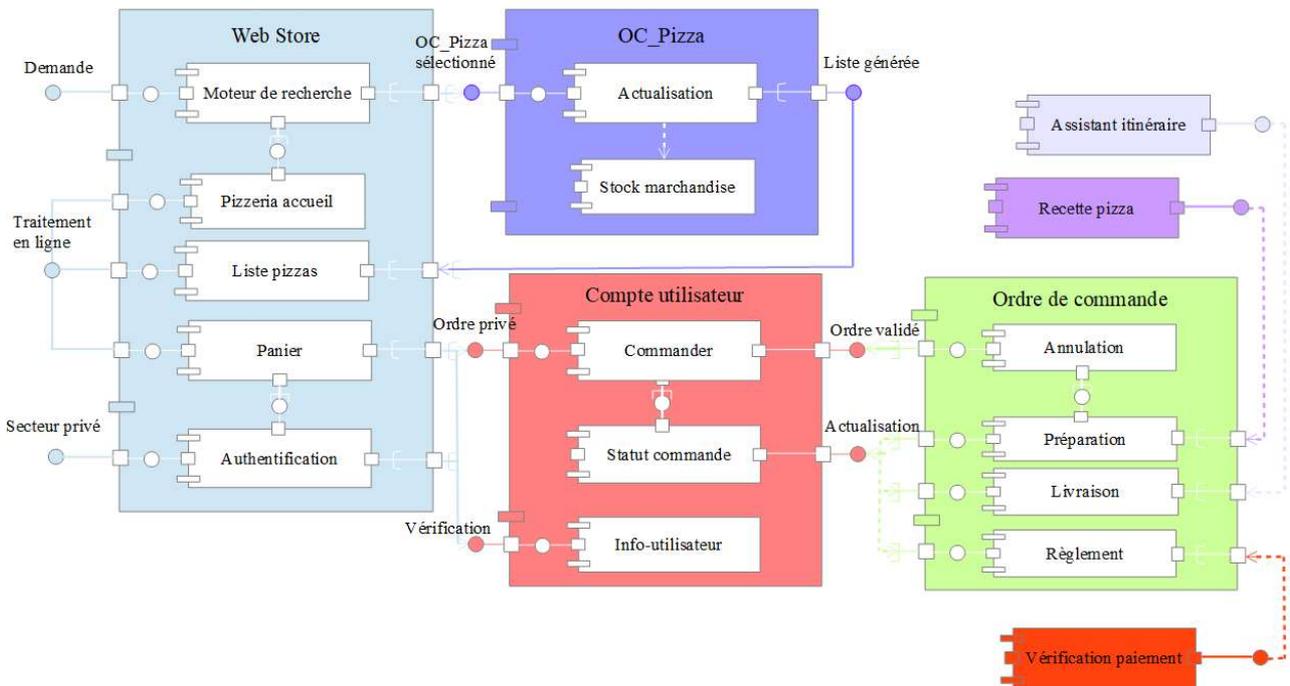


Dans le même principe que tout à l'heure, une pizzeria n'a qu'une adresse et celle-ci n'appartient qu'à une seule pizzeria.

On peut remarquer que chaque attribut à une typologie de donnée. Par exemple le numéro de l'adresse est un « int » qui veut dire « INTEGER » en anglais. Rue, Avenue sont des « char » qui est la traduction de caractère.

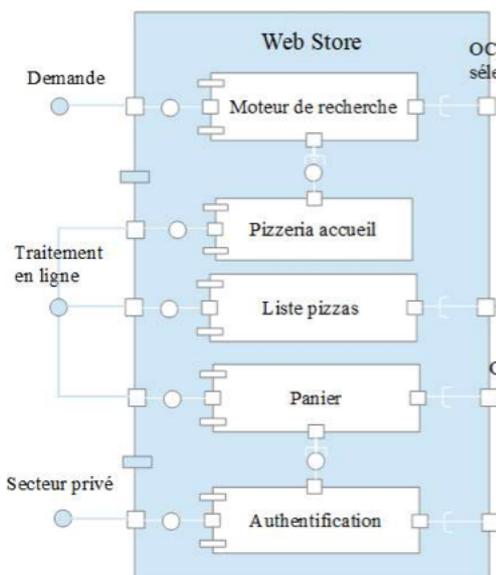
### 3.1.3 Diagramme de composants

DIAGRAMME DES COMPOSANTS



### 3.1.4 Description des composants

#### 3.1.4.1 Le module Web Store



Le moteur de recherche est la première page que l'utilisateur rencontrera. A l'intérieur d'un champ il saisira sa demande et la validera, il en suivra une liste de tout les acteurs répondant à sa demande. Ici l'acteur aura demandé une pizza, ou une pizzeria ou simplement un ingrédient de son choix. Comme OC\_Pizza sera bien référencé il apparaîtra dans le haut de la liste.

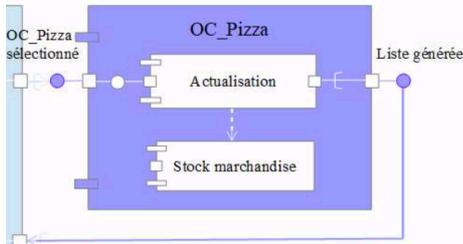
Lorsqu'il vous aura sélectionné vous lui fournirez une page d'accueil présentant vos établissements ainsi qu'un moyen de les localiser. Nous pourrons compléter par une géolocalisation et un itinéraire conseillé. Pendant ce temps le système aura généré une liste de pizza encore disponible au regard du stock de marchandise.

Le consommateur n'aura plus qu'a faire son choix et les intégrer dans son panier.

La validation de sa commande dirigera le client vers un page d'authentification ou, dans le cas où il ne serait pas encore client une page d'enregistrement.

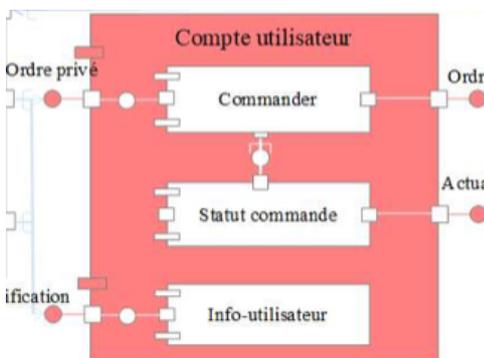


### 3.1.4.2 Le module OC\_Pizza



Le client vient de vous sélectionner et consulte votre page d'accueil. Ici nous organisons la génération de la liste des pizzas en ayant interrogé le stock des ingrédients. Une fois terminé nous la chargeons dès que le client aura cliqué le lien consulter la carte.

### 3.1.4.3 Le module Compte utilisateur

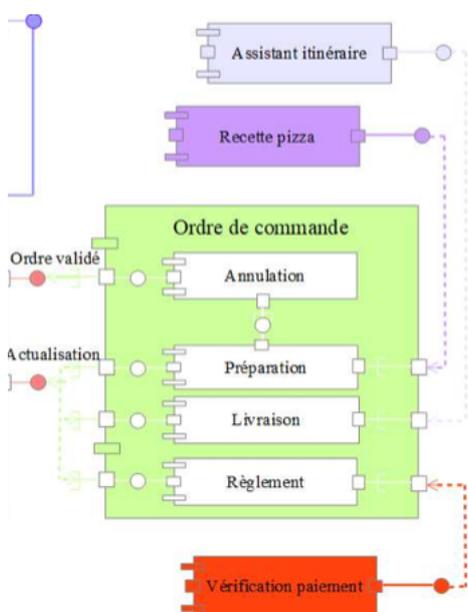


Le client rentre dans la phase de validation de son panier. Pour commander il devra s'être authentifié.

Le statut de la commande tient au courant le client de l'état d'avancement de sa commande.

A tout moment l'utilisateur pourra mettre à jour son compte client afin que ses moyens de contact et de livraison soit garantis.

### 3.1.4.4 Le module Compte utilisateur



Le module assistant itinéraire pourra être activé soit pour diriger le livreur soit pour le consommateur afin de le mener à la pizzeria la plus proche en lui fournissant le chemin le plus rapide.

Le module recette pizza sera un élément intéressant pour fournir au pizzaiolo la liste des ingrédients de la pizza qu'il va préparer mais aussi permettre de la faire évoluer ou d'en créer d'autres.

Le client aura la possibilité de supprimer sa commande tant que celle-ci n'est pas dans la phase de préparation.

Une fois arrivé à ce stade le client sera informé à chaque évolution de la procédure.

L'option livraison si elle est activée générera automatiquement le guide GPS

Tout les modes de règlement sont disponibles et libéreront la commande des pizzas.

Pour les règlements par carte bancaire l'appel à l'organisme vérificateur sera automatiquement activé.



## 3.2 - Application Web

### 3.2.1 Composants HTML 5



**HTML5** (acronyme de Hyper Text Markup Language) est un langage de programmation pour la création de site internet. Il est utilisé pour structurer le document. Il regroupe toutes les parties de saisie et de lien vers d'autres pages du Web ou d'image. Conçue à l'origine pour des pages statiques, il est adossé à d'autres langages pour rendre les sites dynamiques.

pour en savoir plus : <https://fr.wikipedia.org/wiki/HTML5>

<https://www.commentcamarche.net/faq/30122-html-5-qu-est-ce-qui-change>

### 3.2.2 Composants CSS3



**CSS3** (acronyme de Cascading Style Sheets) est un langage de programmation utilisé pour mettre en forme les fichiers HTML. Bien qu'avec des balises à l'intérieur du code on peut les modifier, il est apparu plus judicieux à tous les développeurs de séparer ces deux fonctions. Cela permet de rendre le premier plus lisible et donc plus facile à maintenir à jours. Les fichiers CSS accompagne presque tout les fichier HTML.

pour en savoir plus : [https://fr.wikipedia.org/wiki/Feuilles\\_de\\_style\\_en\\_cascade#CSS3](https://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade#CSS3)

<http://www.css-faciles.com/premiers-pas-css.php>

### 3.2.3 JAVASCRIPT



**JavaScript5** est un langage de programmation de script principalement employé pour rendre les pages HTML interactives, mais aussi pour les serveurs. Il est orienté objet, c'est-à-dire qu'il encapsule un ensemble de propriétés dans un objet, ainsi il est plus facile de transférer un objet qu'un ensemble de données.

pour en savoir plus : <https://fr.wikipedia.org/wiki/JavaScript>

[https://developer.mozilla.org/fr/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/fr/docs/Learn/JavaScript/First_steps/What_is_JavaScript)

### 3.2.3 Sublime Texte



**Sublime Text** est un éditeur de texte disponible sous Mac et Windows. Issu de Vim, celui-ci offre plus de fonctionnalité comme la coloration syntaxique, l'auto complétion et un système de plugins. Il possède aussi des fonctions comme personnalisation des raccourcis la fonction recherche et remplace. Il prend en charge la plupart des langages de programmation.

pour en savoir plus : [https://fr.wikipedia.org/wiki/Sublime\\_Text](https://fr.wikipedia.org/wiki/Sublime_Text)

<https://www.programmez.com/actualites/sublime-text-3-arrive-en-version-finale-26467>



## 3.3 - Applications nomades

### 3.3.1 Android Studio



**Android Studio** est un IDE (acronyme de Integrated Development Environment), on pourrait écrire que c'est un environnement de travail permettant de rédiger, simuler puis créer des applications destinées aux utilisateurs de téléphone autre que iPhone. Il permet d'éditer les fichiers en langages Java/Kotlin et les fichiers de configuration XML.

pour en savoir plus: [https://fr.wikipedia.org/wiki/Android\\_Studio](https://fr.wikipedia.org/wiki/Android_Studio)  
<https://www.frandroid.com/android/>

### 3.3.2 Kotlin sous Android Studio



**Kotlin** est un langage de programmation orienté objet avec un compilateur intégré (cette partie du code est capable de transcrire les lignes d'écriture en ligne interprétable par la machine). Équipé d'un simulateur il est possible de visualiser immédiatement le résultat des modifications apportées à un programme.

pour en savoir plus: [https://fr.wikipedia.org/wiki/Kotlin\\_\(langage\)](https://fr.wikipedia.org/wiki/Kotlin_(langage))  
<https://openclassrooms.com/fr/courses/5353106-initiez-vous-a-kotlin>

### 3.3.3 Swift sous Xcode langage



**Xcode** est un IDE (acronyme: Integrated Development Environment), ou un environnement de développement permettant de concevoir des applications destinées aux utilisateurs iOS, watchOS et tvOS. Il permet d'éditer les fichiers en langages Objective-C, Ruby, Swift.

pour en savoir plus: <https://fr.wikipedia.org/wiki/Xcode>  
[https://www.youtube.com/watch?v=NPdTk\\_PRIyc](https://www.youtube.com/watch?v=NPdTk_PRIyc)

### 3.3.4 Swift sous Xcode langage



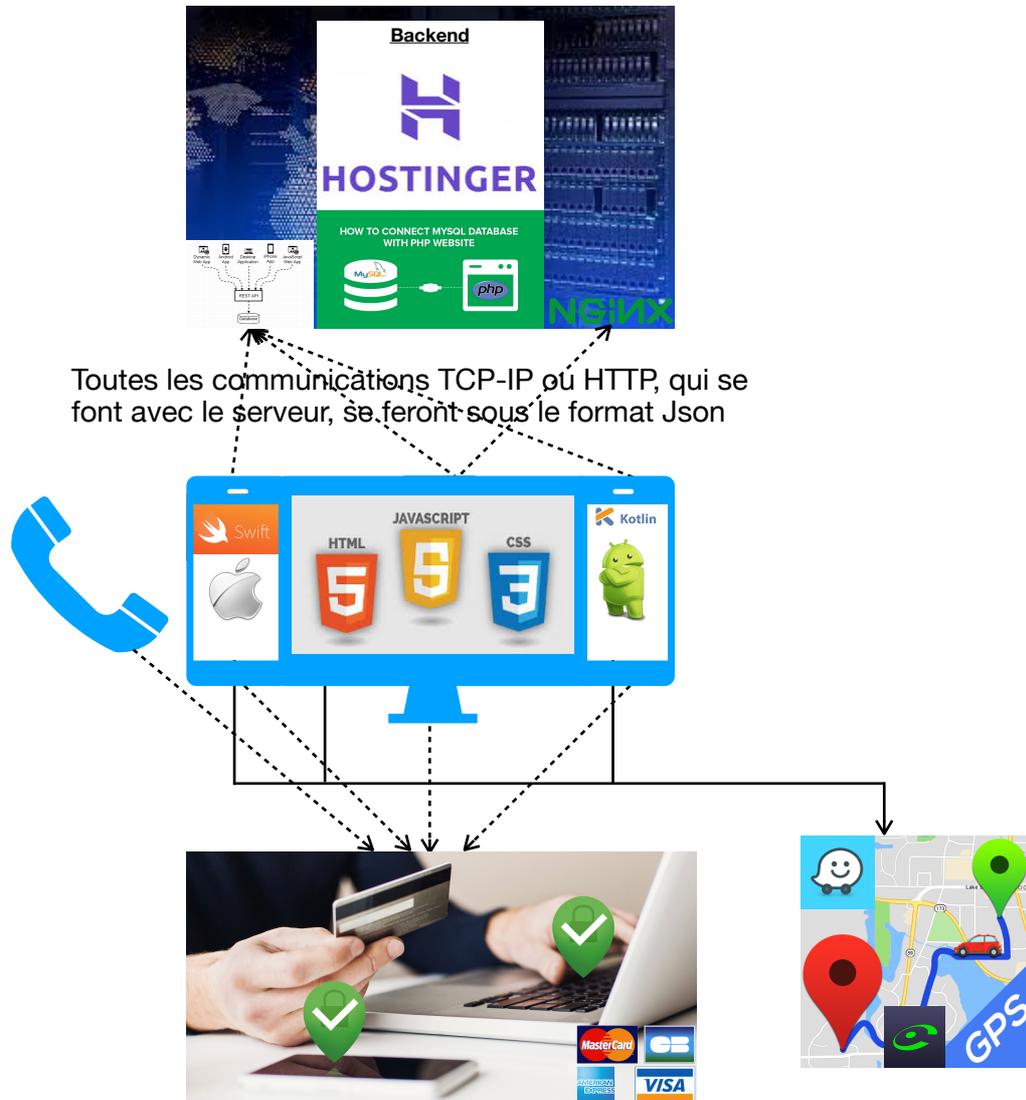
**Swift** est un langage de programmation orienté objet avec un compilateur intégré (cette partie du code est capable de transcrire les lignes d'écriture en ligne interprétable par la machine). Équipé d'un simulateur il est possible de visualiser immédiatement le résultat des modifications apportées à un programme.

pour en savoir plus: <https://fr.wikipedia.org/wiki/Xcode>  
<https://www.apple.com/fr/swift/>



# 4 - Architecture de Déploiement

## 4.1 - Diagramme général de déploiement





## 4.2 - Détail du diagramme

Un **hébergeur web** est une entité, disposant de serveurs maintenu en fonction 24/24. Par une connexion à très haut débit et sous peu que le demandeur y soit habilité, donne accès à des sites, à des logiciels, à une messagerie et à des bases de données conçus et gérés par des tiers.

### 4.2.1 - Serveur de Base de donnée

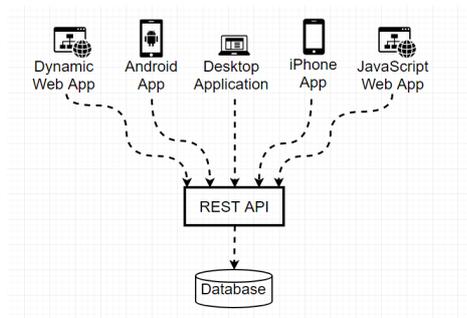


Nous avons retenu l'hébergeur **Hostinger** pour sa pré-disposition à accueillir les bases de données au format SQL . La Structure MAMP embarque le tandem MySQL et PHP idéal pour la conception et l'utilisation d'une BDD Relationnelle.

pour en savoir plus: <https://en.wikipedia.org/wiki/Hostinger>

<https://www.commentcamarche.net/forum/affich-26615994-est-ce-que-quelqu-un-connaît-hostinger-fr>

### 4.2.2 - L'API REST

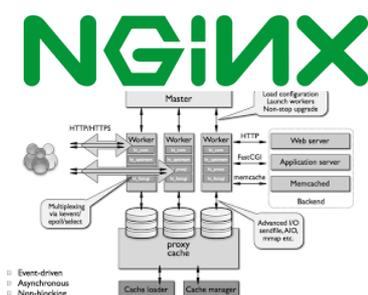


Dans une configuration complète telle que celle attendue, nous devons inclure dans l'attelage du projet l'**API REST** (acronyme de representational state transfer) qui est la référence dans l'interconnexion des différents supports. Elle sera accessible au travers de l'hébergeur.

pour en savoir plus: [https://fr.wikipedia.org/wiki/Representational\\_state\\_transfer](https://fr.wikipedia.org/wiki/Representational_state_transfer)

<https://www.youtube.com/watch?v=ADyVex3AYZE>

### 4.2.3 Le serveur NGINX



**NGINX** est un logiciel pour serveur disposant de capacité de traitement simultané jamais encore égalé.

Le secteur d'activité de bouche les demandes sont principalement concentrées et très nombreuses d'où le choix proposé.

pour en savoir plus: <https://fr.wikipedia.org/wiki/NGINX>



#### 4.2.4 Le Système de base de données

Un **système de gestion de base de données** (acronyme de *SGBD*) est un logiciel système servant à stocker, à manipuler ou gérer, et à partager des informations dans une base de données, en garantissant la qualité, la pérennité et la confidentialité des informations, tout en cachant la complexité des opérations.

pour en savoir plus: <https://fr.wikipedia.org/wiki/MySQL>

<https://www.phpjungle.info/articles/25-mysql-c-est-quoi-a-quoi-ca-peut-me-servir-en-cours.html>

#### 4.2.5 PHP le logiciel d'interface avec MySQL



**PHP** (acronyme de *Hypertext Preprocessor*) est un logiciel orienté objet produisant des pages Web dynamiques via un serveur HTTP. Domicilié sur le serveur il interrogera et renverra des pages Web personnalisées au destinataire des requêtes. Intégré chez l'hébergeur avec la BDD l'utilisation sera simplifier par ma pré-programmation des requêtes.

pour en savoir plus: <https://fr.wikipedia.org/wiki/PHP>

<https://www.php.net/manual/fr/intro-what-is.php>

#### 4.2.5 Apache



**Apache** est le plus répandu des serveurs HTTP sur Internet capable de répondre aux requêtes. Cette position de leader vient par sa compatible avec les principaux acteurs d'OS. Il inclus l'interprétation du langage PHP, Python, serveur proxy, réécriture d'URL .

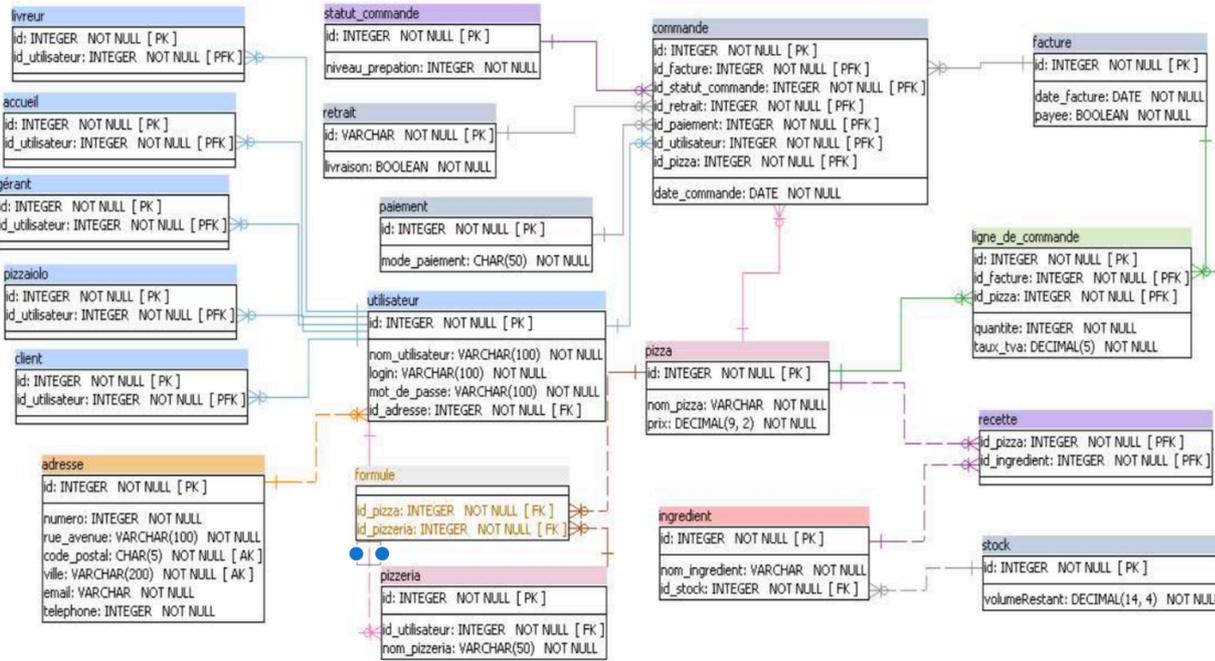
pour en savoir plus: [https://fr.wikipedia.org/wiki/Apache\\_HTTP\\_Server](https://fr.wikipedia.org/wiki/Apache_HTTP_Server)

<https://kinsta.com/fr/base-de-connaissances/qu-est-ce-qu-apache/>



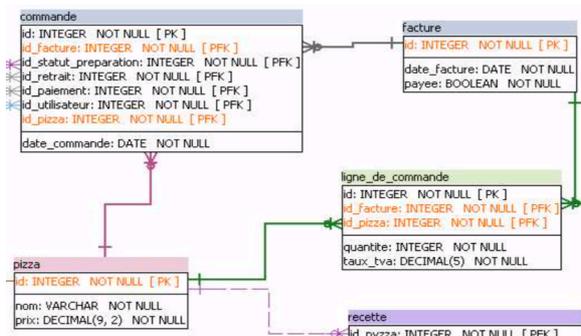
# 5 - Architecture Logicielle

## 5.1 - Principe généraux le MPD



## 5.2 - Détail du MLD

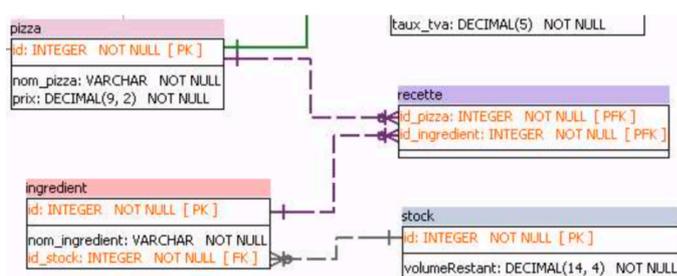
### 5.2.1 Les tables commande, pizza, facture et ligne de commande



La table « ligne\_commande » utilise deux clés primaires étrangères pour relier la table « facture » et la table « pizza ».

Tant qu'à la table « commande » le lien avec la table « facture » est assuré.

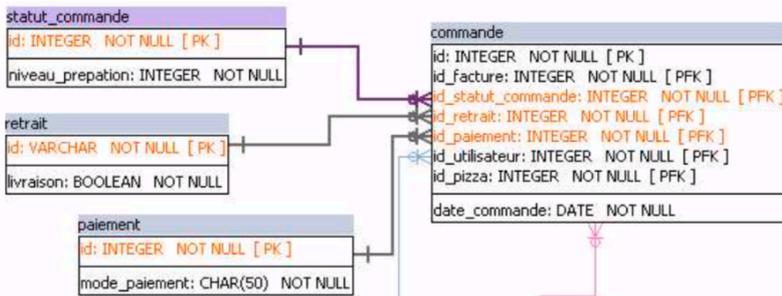
### 5.2.2 Les tables pizza, recette, stock et ingrédient



Dans cette partie nous avons la table recette qui est une table d'association, elle est constituée de deux clés étrangères qui permettront de relier les tables « pizza » et « ingrédient ». Elles serviront d'identifiant . Cette dernière donnera accès au niveau de stock de l'ingrédient. La clé « id\_stock » est une clé étrangère noté [FK] pour Foreign Key

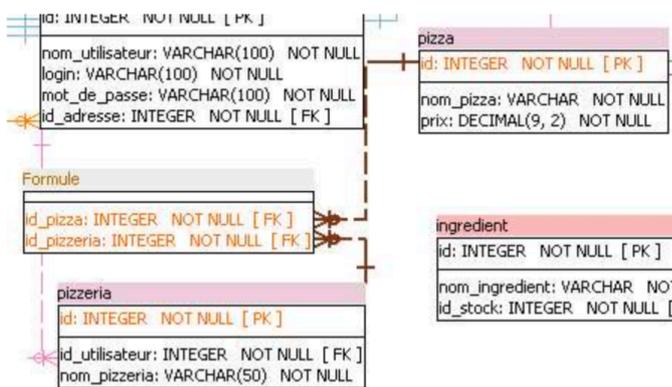


### 5.2.3 Les tables commande, statut commande, retrait, paiement



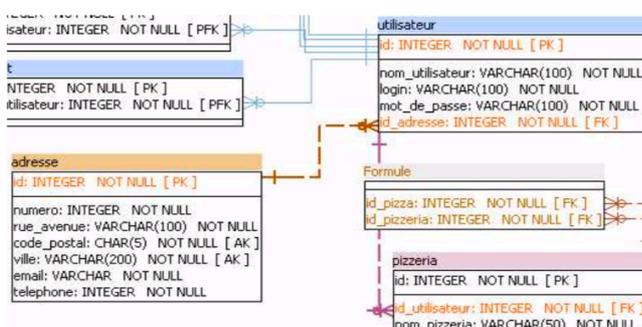
La table « commande » donnera accès à toutes les informations qu'elle contient sous peu que les autres soient équipées d'une clé étrangères  
On peut constater que les liens entre les tables sont des éléments de même type sans cela le lien ne serait pas possible. Ici ce sont des INTEGER. Les attributs de liaison de la table « statut commande », de « retrait », de « paiement » et de « commande » ont été mis en sur-brillance.

### 5.2.4 Les tables pizzeria, formule et pizza



Les tables « pizzeria » transitera par la table « utilisateur » en utilisant la [FK] « id\_utilisateur », et par rebond « id\_adresse » pour atteindre son adresse. Ce procédé est un excellent moyen pour éviter de stocker des doublons, ce qui est souvent un risque de glisser des erreurs de saisie, et donc d'avoir une table moins optimisée. Les attributs nom de la table « pizzeria » est un VARCHAR 'chaîne de caractère'. Ce champs est paramétrable il a été limité à 50 caractères dans ce cas, mais dans la table « adresse » d'autres le sont à 200.

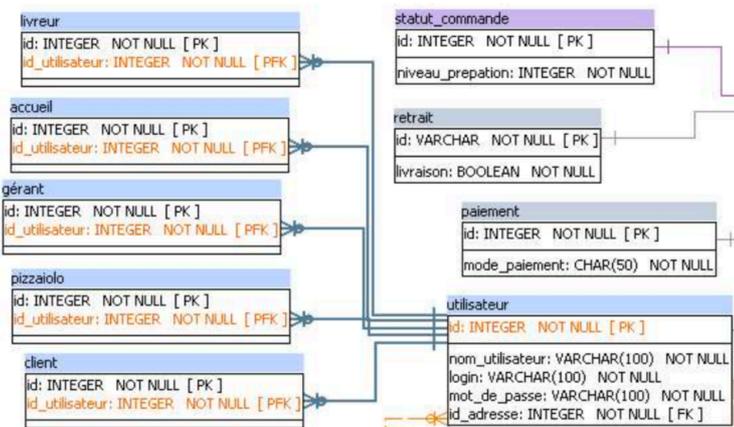
### 5.2.5 Les tables commande, pizza, facture et ligne de commande



Les tables « pizzeria » transitera par la table « utilisateur » en utilisant la [FK] « id\_utilisateur », par rebond « id\_adresse » servira à atteindre l'adresse.  
Les attributs nom de la table « pizzeria » est un VARCHAR 'chaîne de caractère'. Ce champs est paramétrable il a été limité à 50 caractères dans ce cas, mais dans la table « adresse » d'autres le sont à 200.



## 5.2.6 Les tables commande, pizza, facture et ligne de commande



Les tables livreur, accueil, gérant, pizzaiolo et client sont toutes des tables filles de la table « utilisateur », grâce à cela il n'est pas nécessaire de reprendre les attributs de la table « utilisateur » puisque en tant que filles elles héritent de la table mère.

L'héritage est assuré par la clé étrangère [PFK] pour Primary Foreign Key (id\_utilisateur).

C'est un INTEGER 'nombre entier'. Celle-ci nous permettra de nous connecter à la clé primaire « id » [PK] de la table « utilisateur ».

# 6 - Points particuliers

## 6.1 - Gestion des logs

Sauf demande explicite, les logs sont conservés trois mois. Pour des raisons comptables ou autres, il est toujours possible d'étendre ce délai, dans la mesure où ceux que l'archivage soit en adéquation avec les capacités de stockage qui restent à déterminer.

## 6.2 Fichiers de configuration

### 6.2.1 Application web

La configuration du serveur HTTP Apache est effectuée en plaçant des directives dans des fichiers de configuration au format texte. Le fichier de configuration principal se nomme en général httpd.conf. La localisation de ce fichier est définie à la compilation, mais peut être redéfinie à l'aide de l'option de ligne de commande -f. En outre, d'autres fichiers de configuration peuvent être ajoutés à l'aide de la directive Include, et des caractères de remplacement peuvent être utilisés pour inclure de nombreux fichiers de configuration. Des directives de tous types peuvent être placées dans chacun de ces fichiers de configuration. Les modifications dans les fichiers de configuration principaux ne sont prises en compte par httpd que lorsque le serveur est démarré ou redémarré.

Le serveur lit aussi un fichier contenant les types de documents mime; ce fichier est défini par la directive TypesConfig, et se nomme mime.types par défaut.



## 6.2.2 - Datasources

Avec notre Modèle Physique de données PostgreSQL générera automatiquement un fichier « .dump ». Grâce à lui Power Architect construire une base de données vierge constitué de toutes les tables ayant chacune tous les champs que nous avons préparés.

Nous vous avons générer un fichier « OC\_Pizza.dump » permettant de retrouver la base de données remplie telle que nous vous l'avons présenté.

Ce fichier fera partie des annexes.

## 6.3 - Ressources

Le modèle physique de données suivant édité avec le logiciel OpenSource "SQL Power Architect", sera pleinement compatible avec une grande majorité de bases de données. Le fichier généré se trouve en annexe (OC Pizza MPD.pdf)

## 6.4 - Environnement de développement

Le site internet sera développé avec HTML5, CC3, JAVASCRIPT7.

Les applications seront respectivement développé avec Kotlin dans l'environnement de Android Studio pour la version Android et Swift dans l'univers d'Xcode pour celle tournant sous iOS.

La base de données sera générée via SQL Power Architect grâce à notre modèle physique de données.

## 6.5 - Procédure de packaging / livraison

### 6.5.1 Livraison

A la livraison du dossier, le client prendra possession de l'ensemble de la solution.

### 6.5.2 Formation

Si OC Pizza le souhaite un module de formation existe, nous pourrons donc si le besoin s'en fait sentir dispenser les postulas de départ à ceux qui seront amenés à utiliser la BDD sur le serveur.

### 6.5.3 Suivi

La durée de garantie inclus un suivi pendant l'année qui suit la mise en service. Avant la date anniversaire nous vous proposerons de la proroger pour une durée que nous pourrons définir ultérieurement.

Cependant, nous vous proposerons une offre de maintenance dès la mise en service du projet. Celle-ci ne rentre pas dans la typologie des éléments inclus dans l'offre de garantie.



Face à des urgences et des impondérables, notre équipe saura s'organiser pour vous permettre de retrouver au plus vite une situation normale d'utilisation.

---

#### 6.5.4 Problème d'implémentation

Si lors de la livraison de la solution nous détectons un problème d'implémentation, nos services prendront les mesures nécessaires à la remise en conformité sans frais supplémentaires.

---

#### 6.5.5 Mise à jour

Afin d'offrir un suivi de qualité nous nous engageons à vous proposer les mises à jour que notre environnement profitera. Ces opérations périlleuses seront réalisées par notre équipe afin de pérenniser les migrations de bases de données et surtout de son contenu. Toute demande de modification et/ou de développement supplémentaires fera l'objet d'un devis qui restera valide pour une durée d'un mois.