

SOLUTION TECHNIQUE D'UN SYSTÈME DE GESTION DE PIZZERIA :

Suite à la réalisation du dossier de spécifications fonctionnelles ayant permis de dresser les diagrammes de package, de cas d'utilisation et d'activités. On retrouve en bas de page le diagramme de Packages regroupant les acteurs du programmes et les paquets de fonctionnalités. Ce dossier vient détailler les solutions techniques à travers 3 grands diagrammes :

*** 1 Diagramme de classe.....P2 à 3

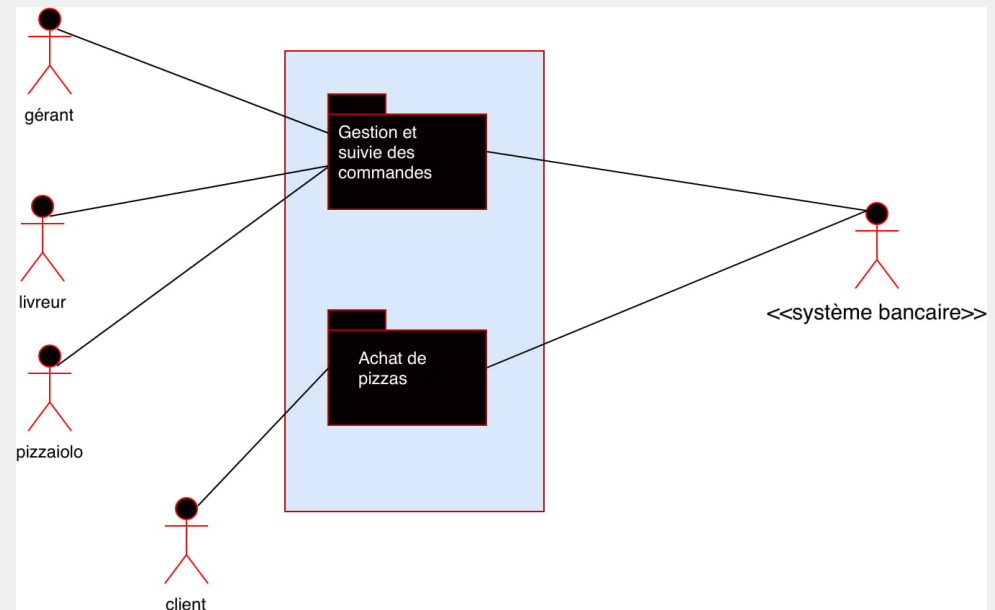
*** 2 Diagramme de composants.....P4 à 5

*** 3 Diagramme de déploiement.....P6 à 7

Le modèle physique de données en fin de document est établie à partir du diagramme de classe, il permet de décrire concrètement et physiquement les tables de la base de données ainsi que leurs relations de clef primaires, étrangères, ainsi que les index détaillés.

*** 4 Modèle physique de données.....P8 à 9

Ces diagrammes permettent de "conceptualiser" et de préparer la futur base de données qui sera le socle du serveur et du Back-end; le même qui alimentera le Front-End visibles aux clients.



1 Domaine fonctionnel du système : Classes et instances du domaine fonctionnel

*** Présentation :

le diagramme de classe : ce diagramme représente toutes les entités internes ou externes à votre programme. On appelle toute ces entités des "classes". Ici, votre schéma contient une quinzaine de classe, répartie en 4 grandes catégories de classes que sont : La pizzeria, le système de paiement, Le stock de pizza, et la gestions commande/clients.

*** Relations :

on retrouve entre vos classes 3 grands types de relations décrivant les interactions et échanges entre ces classes. des relations de composition, d'agrégation et de multiplicités.

* Pour exemple, ici la facture dépend du système de commande et ne peut être présente sans commande, vos commandes seront donc composés de factures. Ces mêmes factures sont en relations avec l'email et le client. De ce fait, toutes ses données sont donc dépendantes des informations saisie par le client lors de la commande. La facture est le résultat, la composition de ces autres classes.

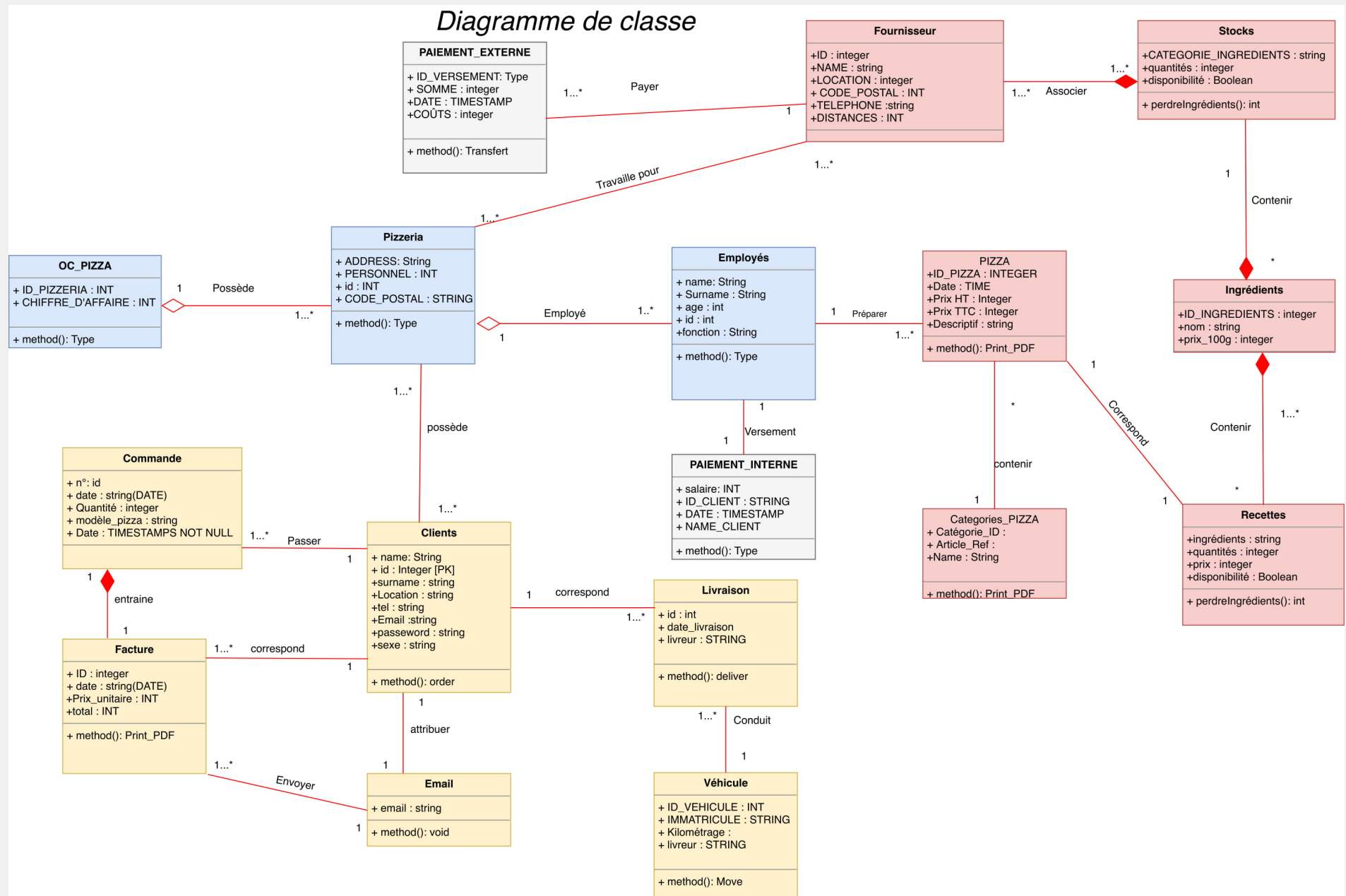
* Par ailleurs, on voit que la gestion des stocks est découpée en 3 entités "recettes", "ingrédients" et "stocks". la classe stock aura pour attributs des quantités, une disponibilité sous forme de vrai ou faux (aussi appelé Booléen) mais aussi une catégorie d'ingrédients. Un fournisseur sera donc associer à votre stock et pourra prendre en charge une ou plusieurs catégories d'ingrédients.

* Le système de paiement, est lui divisé en 2 classes externe et une interne. on retrouve comme dans chaque classe un ID (ici de paiement). Mais aussi une Date, un entier réel indiquant le montant du versements. Les paiements externes sont donc ici liées à vos fournisseurs. Tandis que les paiements internes sont liées à vos clients. On retrouve un ID en clef étrangère ce qui permet de référencer avec exactitude la colonne du receveur dans la table paiement

* Les pizzas sont associées chacune à des catégories permettant de plus vite effectuer les requêtes et recherches au sein de la future base de données. Ces mêmes pizzas seront préparés par un seul employés par pizzeria, cet employé lui peut travailler pour toutes les pizzeria du groupe OC PIZZA mais est assigné à une.

A partir de ces relations entre classes permettant de poser les bases de relations entre les classes, j'ai crée un modèle physique et concret de données représentant la base de données finale. Aussi appelé MPD, en fin de document.

Diagramme de classe



2 Composants du système : Interaction des composants internes et externes du programme

*** Introduction :

Le diagramme de composants permet d'identifier visuellement les différents composants concrets, techniques internes et externes à votre site web de pizzeria. Ils contiennent par exemple les exécutables, les bibliothèques, les fichiers... Mais aussi des données comme la base de données. On met ici en évidence les dépendances (Quel composant utilise quel composant ?)

Les composants fournissent des interfaces; pour fonctionner ils ont besoin d'autres composants, on appelle "interface fournie" (représentées par un cercle) et "interface requise" (représentées par un demi-cercle) ces connecteurs d'assemblages.

Enfin, les carrées représentées aux extrémités des composants (aussi appelées "ports") servent de point de connexion aux connecteurs d'assemblages définis précédemment

*** Relations :

* Les composants sont répertoriés en 2 grands systèmes de composants : 1 les composants tiers et 2 les programmes serveurs. Ces grands paquets interagissent ensemble et s'échangent des informations dynamiquement pour les fournir ensuite à l'utilisateur. En effet le serveur et le Back-end (c'est-à-dire la surface invisible mais nécessaire au fonctionnement de votre programme, fournissant l'interaction entre les données entrantes et sortantes) sont contenus dans un grand composant "Serveur".

* De l'autre côté on retrouve le navigateur clients affichant les informations après interprétation du code HTML fournie par le Back End. L'API Google maps et du paiement font aussi partie de cet ensemble car ils sont extérieurs au système principal. C'est donc ces API qui vont fournir des informations comme des autorisations de paiements ou bien des coordonnées géographiques.

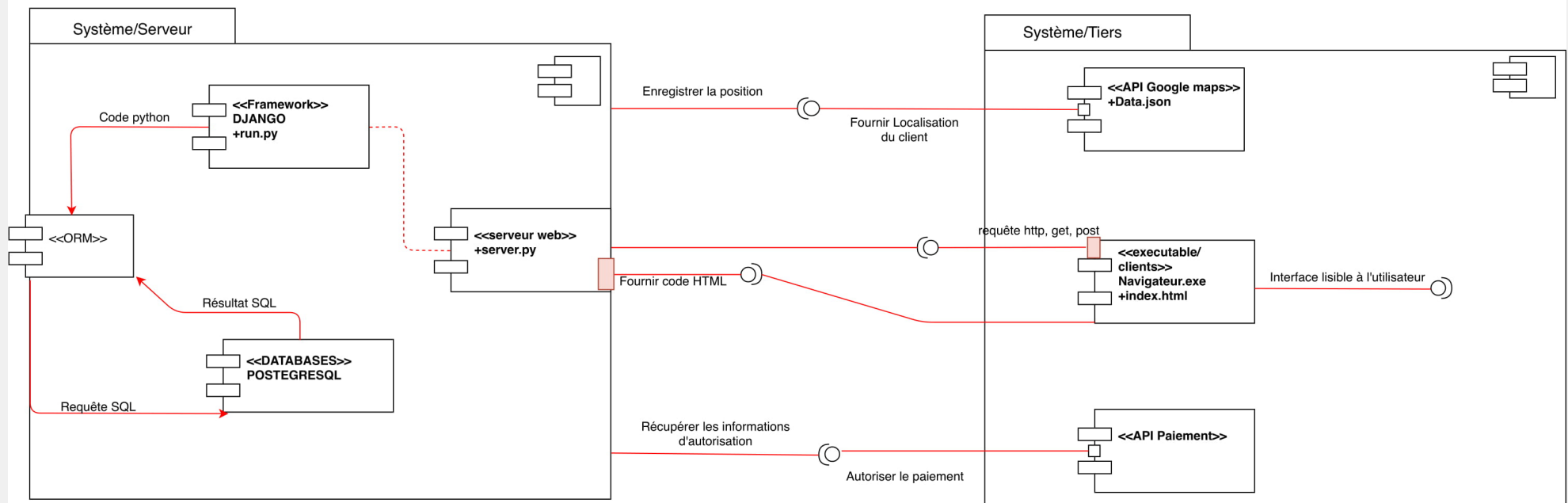
* Techniquement votre base de données est gérée par un ORM propre au Framework Django du langage Python. Cet ORM (object-relational mapping) simule une base de données relationnelles orientées objet, comme vue précédemment dans le diagramme de classe. La base de données POSTGRESQL communique avec L'ORM

Ces composants ici détaillés en détail sont de nouveaux présents dans le déploiement de ces composants expliqué dans la page suivante.

Diagramme de composant

Diagramme de composants

V2



3 Déploiement du système : Organisation physique des composants, déploiement détaillé

*** Rôles :

Le diagramme de déploiement fait parti des diagrammes structuraux (statique), il représente :

- * la disposition physique des ressources matérielles qui constituent le système et montre la répartition des composants (élément logiciels) sur ces matériels.
- * La nature des connexions de communication entre les différentes ressources matérielles.

*** Diagramme de Composants/Déploiement :

Tandis que les composants mettent en évidence les relations de dépendances, le déploiement suivant identifie aussi des éléments de la réalité physique comme un ordinateur, une imprimante, un modem, un serveur...

*** Introduction :

Le diagramme de déploiement est représenté ici par 5 "noeuds". Ces noeuds sont des classeurs ayant la forme de boîtes en 3 dimensions contenant les composants régissant le programme.

On décrit donc ici le déploiement physique des informations générées par le Logiciel Principal OC PIZZA, ces informations sont appelées artefacts dans le jargon.

*** Relations :

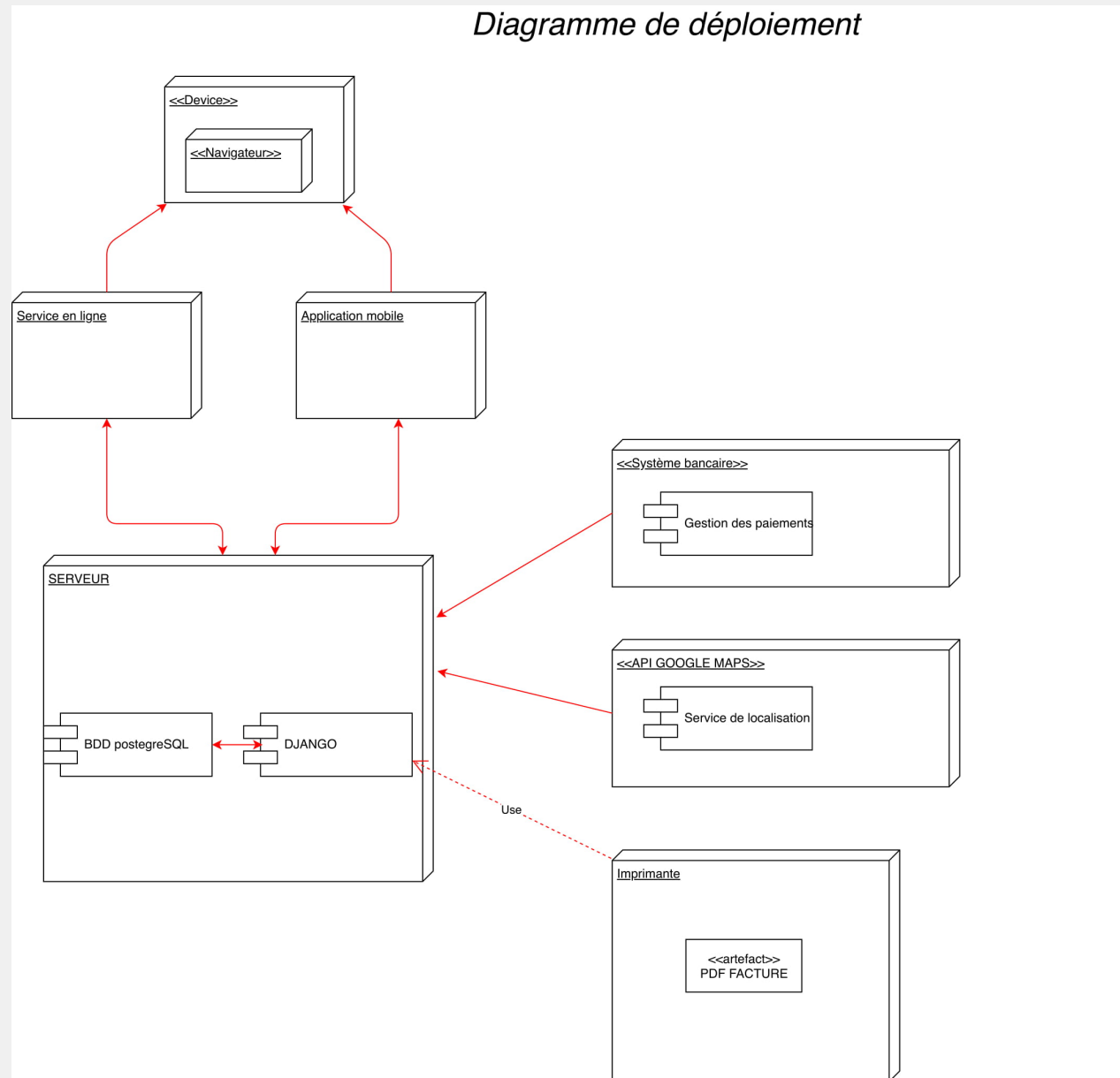
* Au sein de ces diagrammes, les lignes d'associations représentées par des flèches représentent l'échange d'information entre le back end et le site web en ligne/l'application mobile.

* Le système bancaire et l'API fournissent eux des autorisations et des informations géographiques, votre site web sera donc dépendant de ces composants illustrés ici par les pointillets.

* L'espace utilisateur contient lui une sous-boîte représentant le navigateur et plus spécifiquement le "décodage" graphique du HTML5 pour afficher des informations compréhensibles à l'utilisateur. L'échange d'informations allant dans les deux sens c'est ici une double flèche qui caractérise ce même échange.

* L'imprimante ici interagira directement avec le programme Back end fournissant automatiquement le PDF imprimable de la commande du client.

Diagramme de déploiement



4 Modèle relationnel de la base de données : interaction entre les tables

Le modèle de données présent ici découle du diagramme de classe. sa création aboutie à l'implémentation d'une base de données relationnelle au sein d'un SGBDR.

*** Composants utilisées :

Afin de générer en langage SQL la MPD, on peut utiliser un atelier de génie logiciel, ici je me suis servie de SQL power architect pour dessiner la structure et générer un code en Langage MySQL.

*** Clef primaires :

Les clefs primaires vont permettre d'identifier de manière unique un ensemble de données.

J'ai conçu les clefs primaires en respectant deux critères : la constance et l'unicité. A partir de ces critères on crée une clef primaire composite ou bien à l'aide d'un attribut ID dans une nouvelle colonne. On note la dénomination [PK]

*** Clefs étrangères :

Les clefs étrangères découlent donc des relations de multiplicités illustrées plus tôt ! En effet le livreur conduit un scooter, on retrouve donc l'ID (clef primaire) de notre classe livraison en clef étrangère de notre classe Véhicule en guise de conducteur.

La relation entre la table Pizza et Employé est légèrement différente, un employé peut préparer plusieurs pizzas mais une pizza ne peut être préparée que par un pizzaiolo. On retrouvera alors l'ID d'un pizzaiolo en clef étrangère sur plusieurs lignes de notre table pizza.

*** Le rôle des index :

Afin d'optimiser la recherche dans la BDD l'index pointe directement les lignes de la table à l'aide d'un critère.

J'ai créé des index notamment sur les tables contenant des codes postaux et des noms en spécifiant un ordre croissant en ajoutant une caractéristique "clustered" sur ces index. Si les colonnes forment un groupe, on peut paramétrer l'index comme étant unique. On note la dénomination [AK].

*** DUMP

Le dump en fichier joint est une sauvegarde des instructions SQL de l'insertion des données et de la constitution des tables.

Modèle physique de données

