

Spécifications techniques

Sommaire:

I.	<u>Etude du contexte.....</u>	<u>2</u>
II.	<u>Etude des packages.....</u>	<u>3</u>
III.	<u>Etude du cycle de vie d'une commande.....</u>	<u>4</u>
IV.	<u>Description du domaine fonctionnel.....</u>	<u>7</u>
V.	<u>Etudes techniques.....</u>	<u>12</u>

Etude du contexte

En étudiant le cahier des charges on peut distinguer les différents acteurs qui interagissent avec le système de la pizzeria.

Voici la liste :

- ❖ Le client
- ❖ Le préparateur de commande
- ❖ Le Livreur
- ❖ Le manager
- ❖ Le système bancaire
- ❖ La base données

Acteurs principaux

Le client pourra passer une commande via le site internet de la pizzeria qui proposera au client un large choix de produits en fonction du stock disponible. Il aura la possibilité de payer en ligne ou en espèce à la livraison du produit.

Les préparateurs de commande utilisent le système pour suivre les commandes validé en attente de préparation ainsi que la disponibilité de la matière première dans le stock, ils pourront aussi mettre à jour l'état des commandes en cours.

Les livreurs pourront récupérer les coordonnées des dernières commande en attente de livraison et livrer le produit au client.

Le manager pourra accéder au chiffre d'affaires, ajouter ou modifier un produit dans le catalogue et lire les messages d'erreur provenant du système.

Acteurs secondaire

Le système bancaire pour payer en ligne.

La base de données est utilisé pour stocker des informations concernant la pizzeria et les utilisateurs.

Etude des packages

Le système peut-être divisé en trois parties

I. Interface d'achat

Le package "interface d'achat" regroupe la partie site web qui permettra au client de commander des produits en ligne. Il pourra visualiser une liste de produits, créer un panier pour stocker tous ses articles, de proposer divers moyens de paiement et de vérifier l'avancée de sa commande.

II. Interface d'administration

Le package "interface d'administration" regroupe les fonctionnalités de gestions administrateur que propose le site internet de la pizzeria

II. Interface des utilisateurs

Le package "Interface des utilisateurs" s'occupe de la collecte des informations du client et de l'identification des différents types d'acteurs pouvant communiquer avec le site internet.

Acteur	Interface d'achat	Interface d'administration	Interface des utilisateurs
Client / Visiteur	X		X
Préparateur de commande		X	
Livreur		X	
manager		X	
système bancaire	X		
base de données	X	X	X

Etude du cycle de vie d'une commande

Le cycle de vie d'une commande se construit en plusieurs étapes, allant du choix des produits par le client jusqu'à la livraison. Elle peut-être supprimée ou modifiée par l'ordre du client avant la validation du panier de commande.

Statuts d'une commande

Une commande peut avoir quatre statuts :

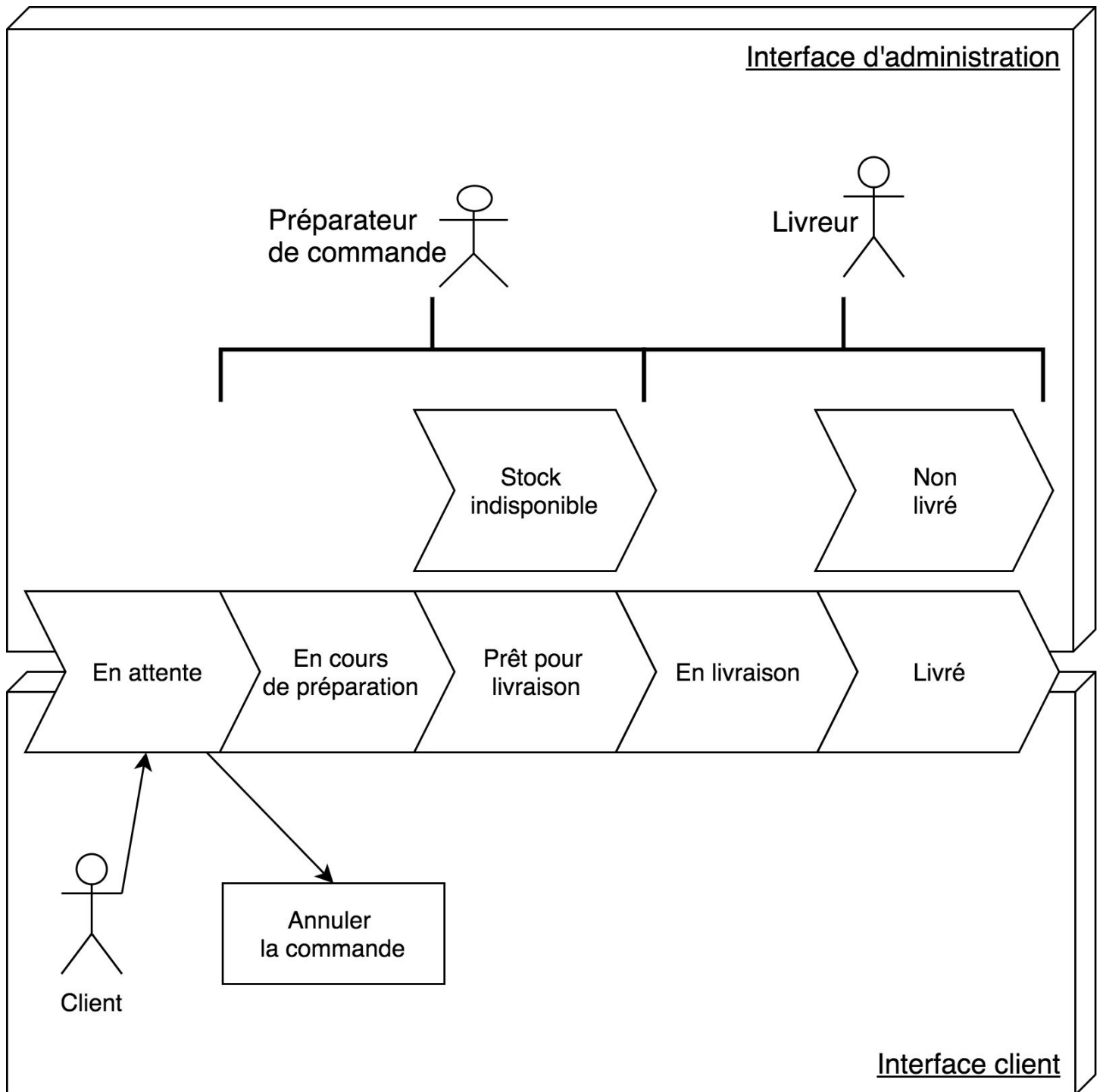
- Statut «en attente» :
Ce statut est généré par l'action du client «Valider son panier ».
C'est aussi le statut de création de la commande. Ce statut veut dire que la commande est passée au système mais n'a pas encore été prise en charge par un préparateur de commande.
- Statut « en cours de préparation » :
Ce statut est généré par l'action d'un préparateur de commande, lorsqu'il commence la préparation de la commande.
 - statut « stock indisponible » :
Lors de la préparation de la commande, le préparateur a rencontré un problème avec le stock d'ingrédient et a dû annuler la commande.
- Statut «en livraison» :
Ce statut est généré par l'action d'un livreur lorsqu'il reçoit la commande préparée.
- Statut «Livré» :
Pour que ce statut soit atteint il faut que la commande soit livrée et payée. C'est le livreur qui prend en charge cette étape final et la mise à jour du statut.
 - Statut «Non livré» :
Le livreur a reçu une mauvaise adresse et le client est injoignable.

Annulation d'une commande

Après validation du panier, l'annulation d'une commande peut encore être faite par l'action du client : elle n'est possible que lorsque la commande est en statut « en attente ». Dans ce cas, le système supprime la commande de la base de données et rembourse le client si il a payé en ligne.

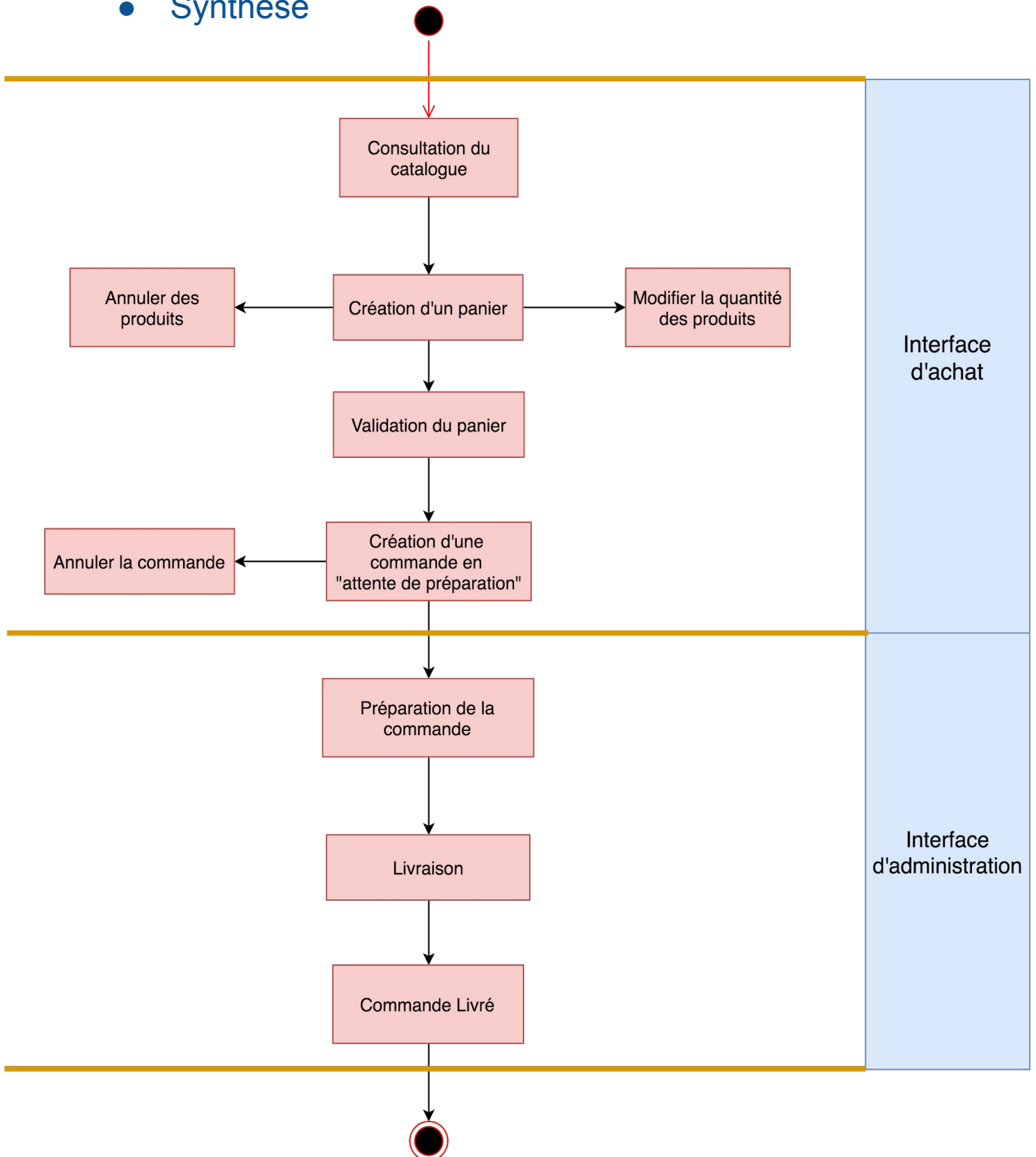
Etude du cycle de vie d'une commande

- Synthèse



Etude du cycle de vie d'une commande

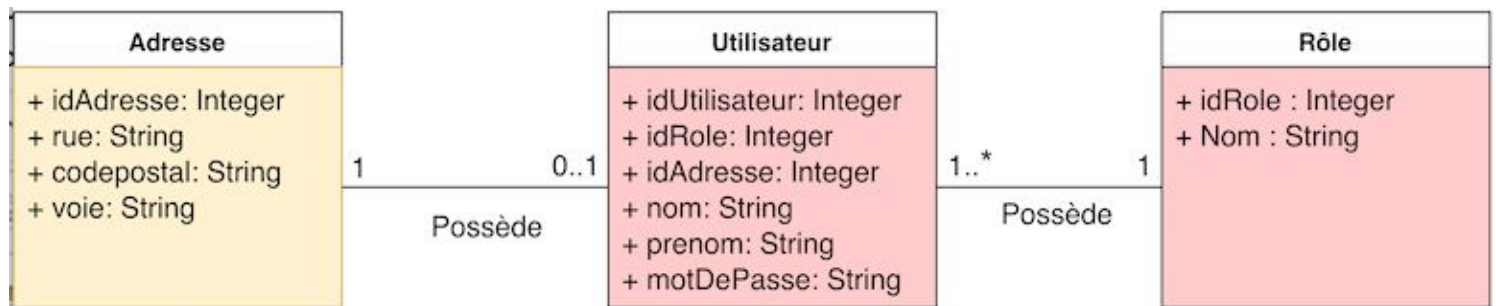
- Synthèse



DESCRIPTION DU DOMAINE FONCTIONNEL :

Dans cette partie, nous expliciterons les différentes entités du domaine fonctionnel permettant la construction d'un modèle physique de données.

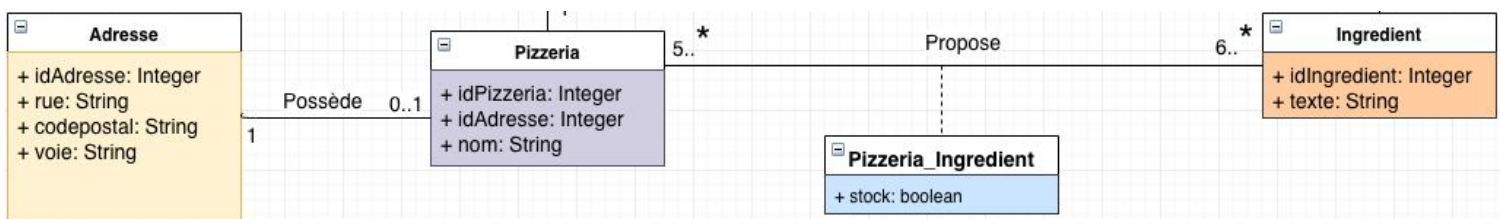
L'utilisateur:



Un **utilisateur** possède une et une seul **adresse** et une **adresse** peut être détenue par aucun ou un **utilisateur**.

Un **utilisateur** possède un et un seul **rôle** et un **rôle** doit être détenue par au moins un **utilisateur**.

Les pizzerias:

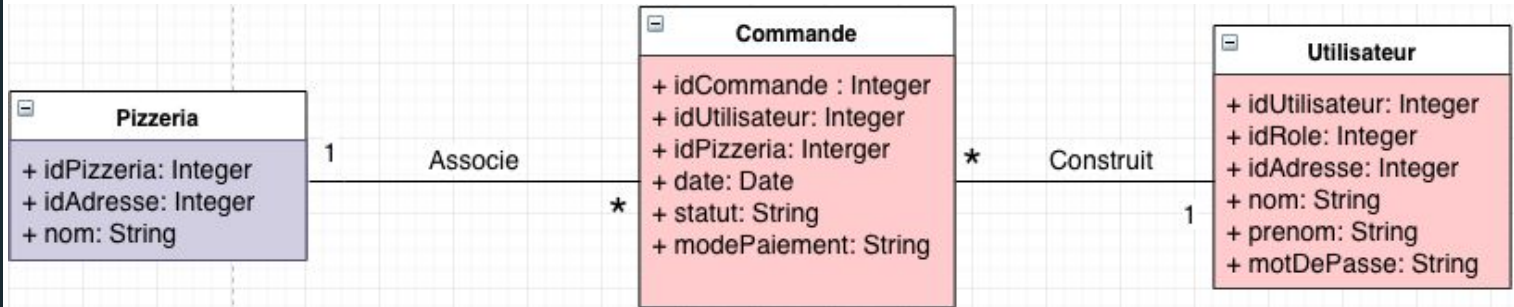


Une **pizzeria** possède une et une seul **adresse** et une **adresse** peut être détenue par aucune ou une **pizzeria**.

Une **pizzeria** doit proposer au moins six un **ingrédient** et un **ingrédient** doit être proposé par au moins cinq **pizzerias**.

DESCRIPTION DU DOMAINE FONCTIONNEL (suite) :

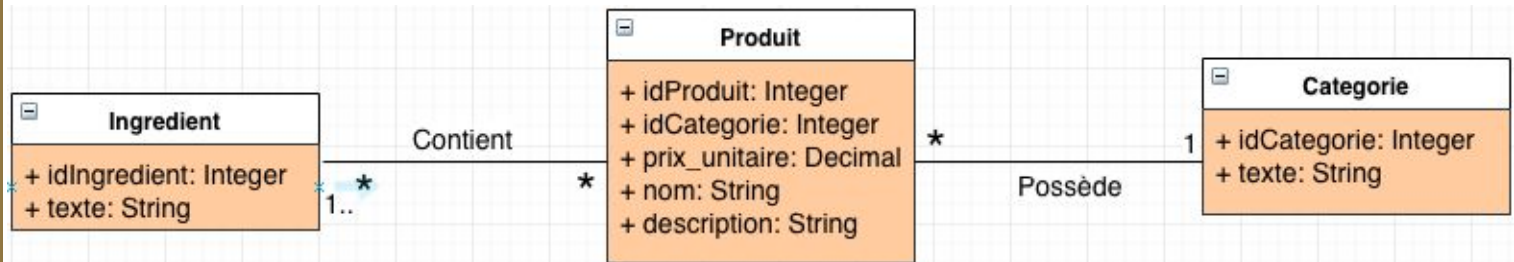
Les commandes:



Une **commande** est associée à une et une seule **pizzeria** et une **pizzeria** peut être associée à aucune ou plusieurs **commandes**.

Une **commande** est construite par un et un seul **utilisateur** et un **utilisateur** peut construire aucune ou plusieurs **commandes**.

Les produits:

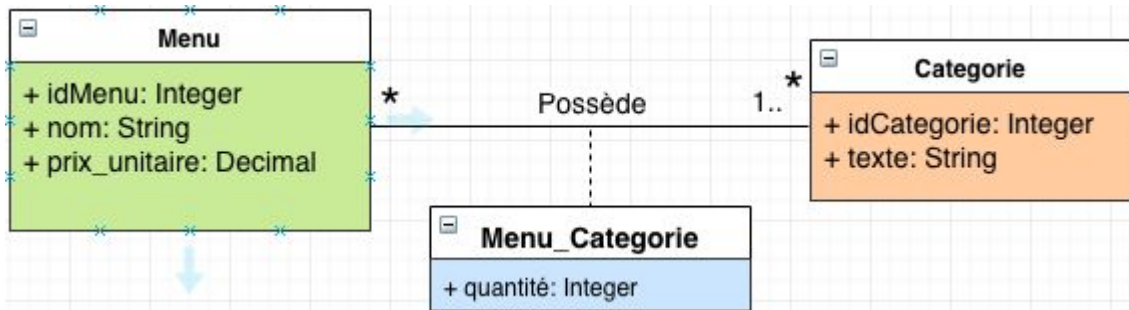


Un **produit** contient au moins un **ingrédient** et un **ingrédient** est contenu dans aucun ou plusieurs **produits**.

Un **produit** possède une et une seule **catégorie** et une **catégorie** est détenue par aucun ou plusieurs **produits**.

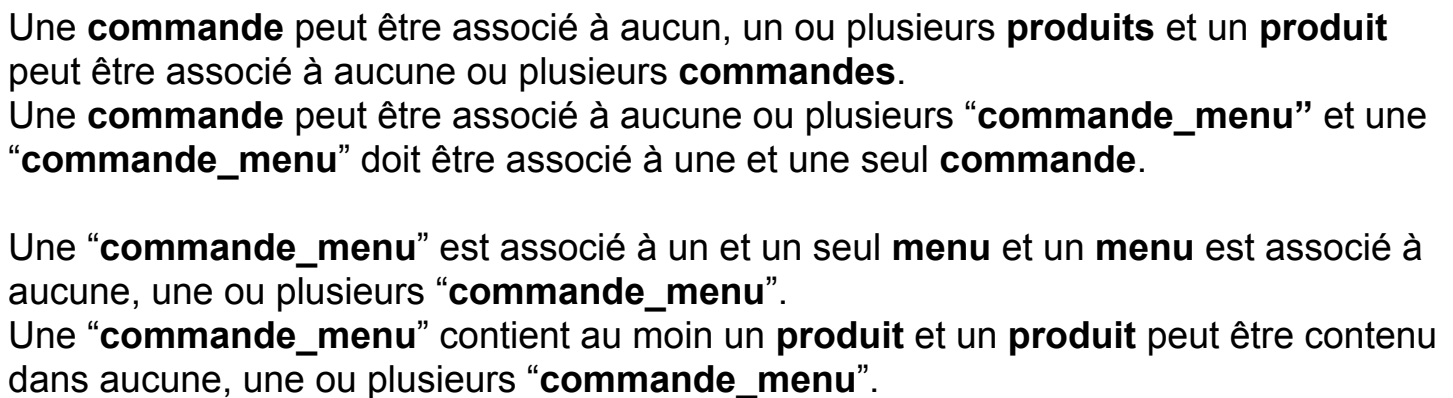
DESCRIPTION DU DOMAINE FONCTIONNEL (suite) :

Les menus:

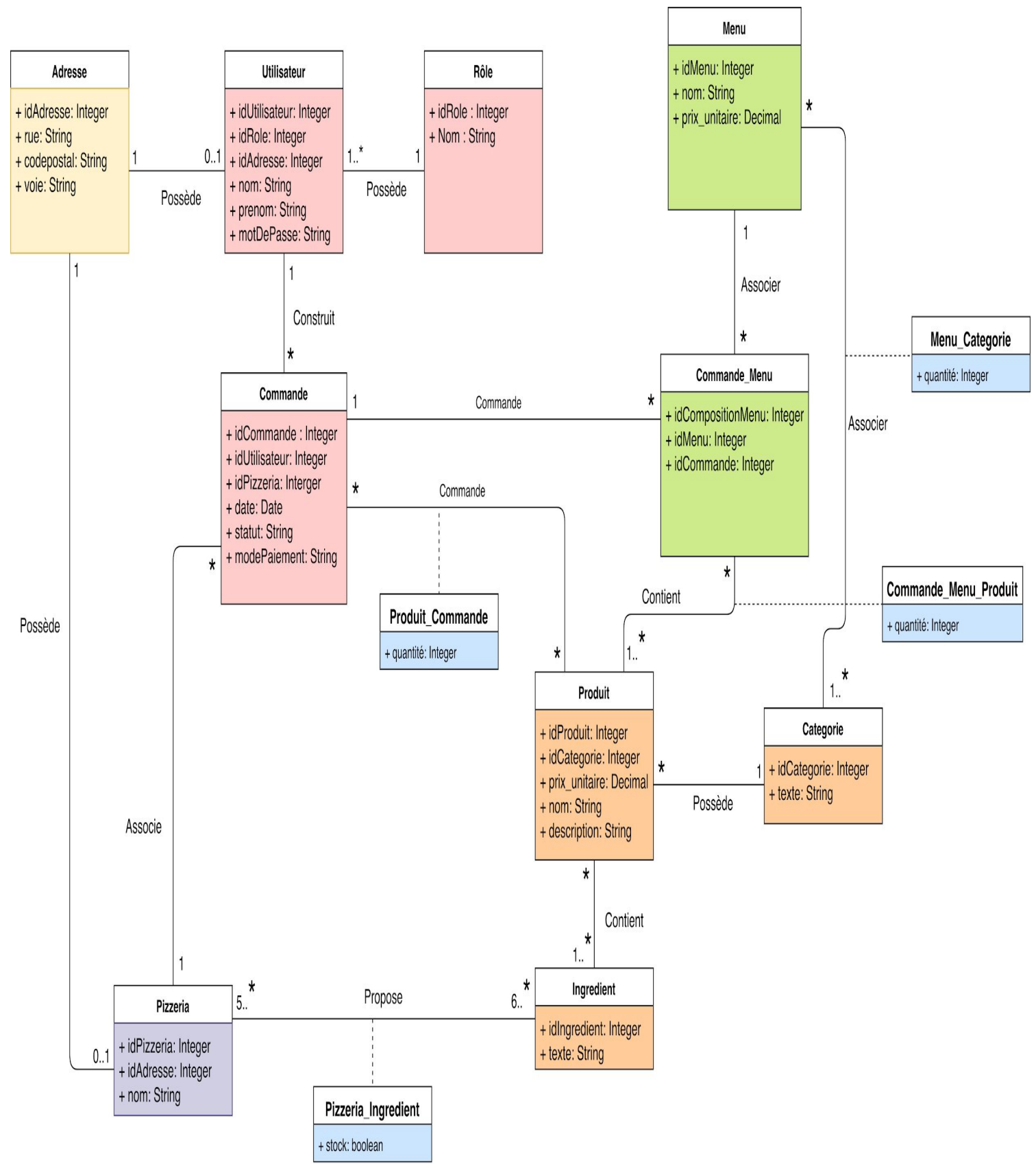


Un **menu** est associé à au moins une **catégorie** et une **catégorie** peut être associé à aucun ou plusieurs **menus**.

Relation entre les commandes et les produits et entre les commandes et les menus:



Synthèse des objets retenus dans un diagramme de classe modélisant le domaine fonctionnel :



Etudes techniques

Pour une mise en œuvre fiable, efficace et évolutive des besoins clients décrit ci dessus, il est proposé d'utiliser un ensemble de technologies web parmi les plus fiable et au développement très actif.

Django framework

Django est un framework web libre et open-source, écrit en Python facilitant la création de sites Web complexes, reposant sur des bases de données. Django met l'accent sur la rapidité de développement d'une application.

Mysql

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public que par des professionnels.

PayPal

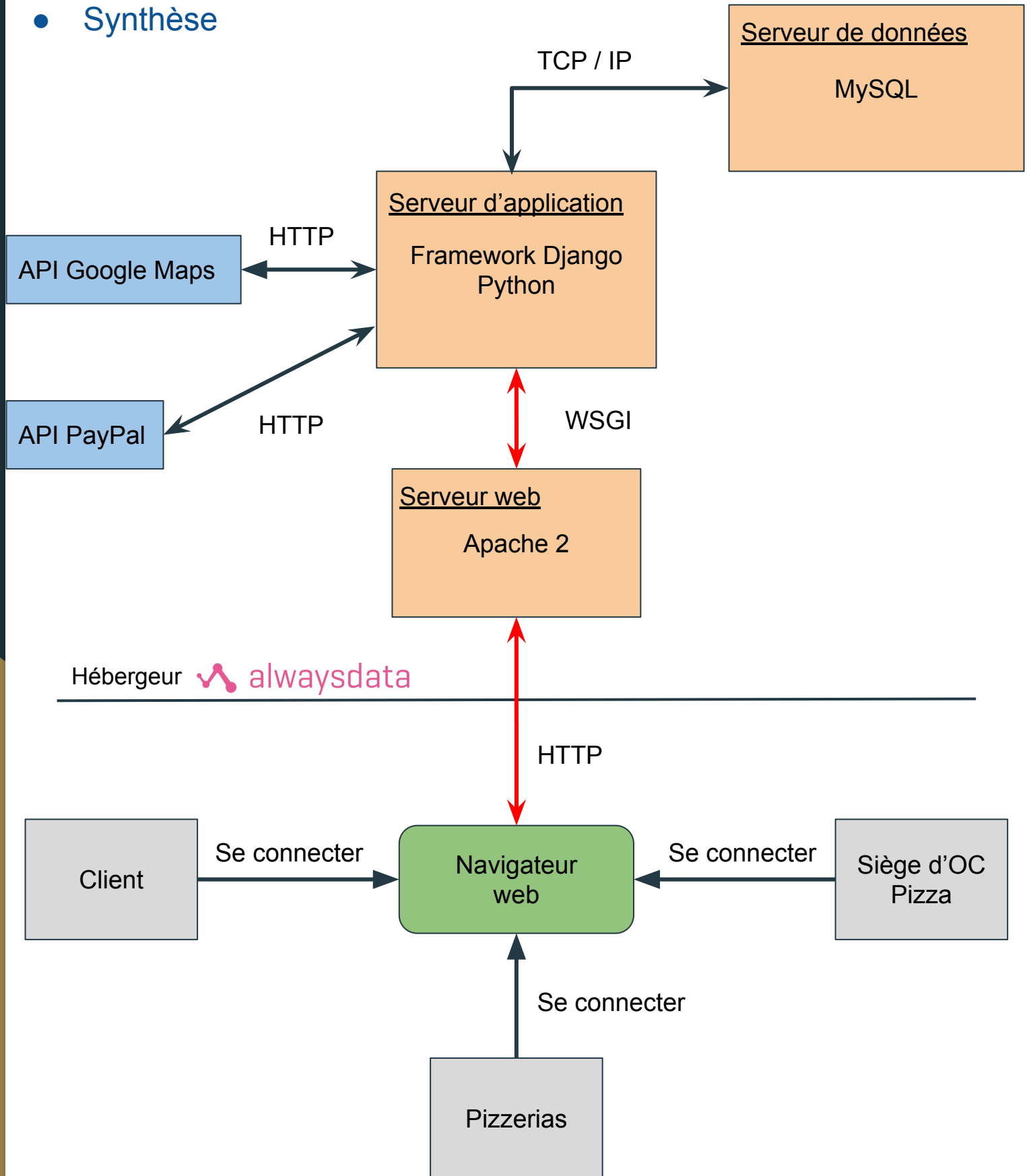
PayPal est un service de paiement en ligne qui permet de payer des achats sans avoir à recommuniquer ses coordonnées bancaires, une adresse de courrier électronique et un mot de passe étant suffisants. Paypal est accepté comme moyen de paiement par plus de 16 millions de commerçants dans le monde.

Google MAP API Distance Matrix API

LA Distance Matrix API est un service qui fournit la distance et le temps pour une matrice d'origines et de destinations. L'API renvoie des informations basées sur l'API Google Map.

Schéma de déploiement

- Synthèse



Les composant interne

- Explication

Serveur d'application :

→ Nous utiliserons un serveur programmé en Python en respectant les bonnes pratiques de la PEP 8 et utilisant le framework Django. Toute la logique métier se trouvera sur ce même serveur.

→ Pour écrire les pages HTML, nous utiliserons jQuery, AJAX, CSS et JavaScript :

- CSS permettra de modifier facilement la présentation des pages web demandées par l'utilisateur.
- JavaScript pour rendre les pages web plus fluides, ergonomiques et interactive (compatible avec tous les supports numériques).
- AJAX permet de mettre à jour une partie seulement de la page web en limitant les échanges client/serveur (sur certaines requêtes, l'utilisateur peut obtenir une réponse très rapide).

Accès aux données :

Utilisation du pilote mysql-connector-python.

Serveur web :

Nous utiliserons Apache 2 (opensource) comme serveur HTTP.

→ Le protocole de communication entre notre projet Django et le serveur Apache est WSGI. Ceci nous permettra de gérer notre application web en conservant les avantages de la programmation objet de Python. Ainsi les requêtes HTTP sont transformées en objets Python lors d'un appel au serveur d'application et inversement.

Serveur de données :

→ Nous utiliserons le système de gestion de base de données relationnelles MySQL.

→ La base de données portera le nom de « oc_pizza » (unique base de données du projet).

Les composant externes

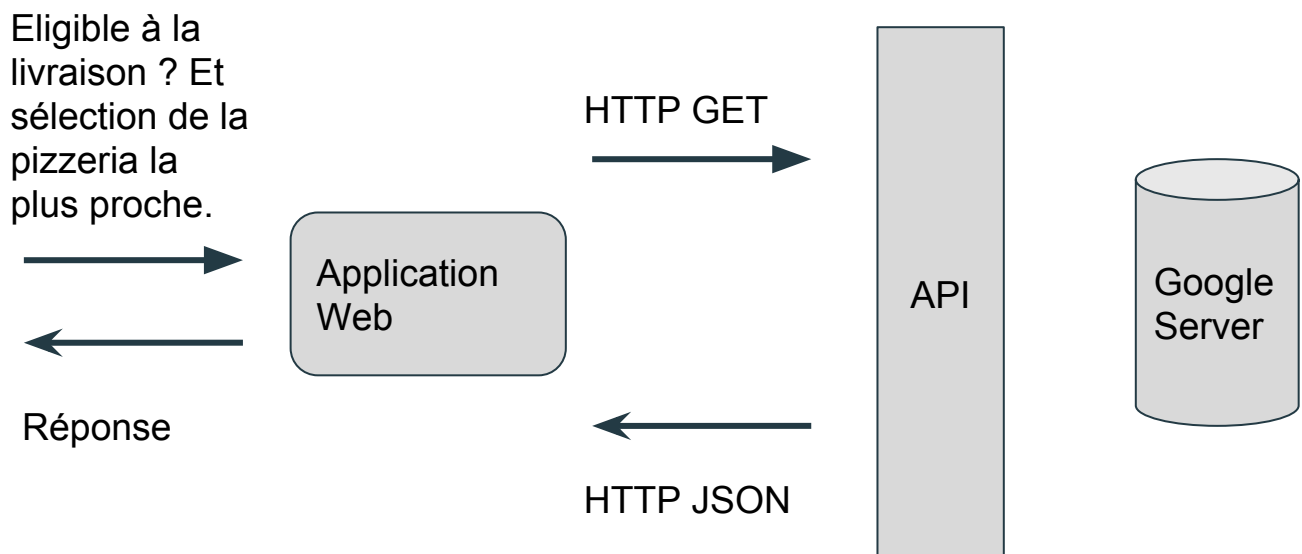
- Explication d'utilité

Google Maps Distance Matrix API:

Afin de localiser la position du client par rapport aux différentes pizzeria, l'application se doit de communiquer avec une API de géolocalisation. Nous avons sélectionné Google Maps Distance Matrix API.

L'idée serait d'envoyer une requête HTTP contenant l'adresse du client et les différentes adresses des pizzerias à l'API. Nous attendons en retour des distances calculée par l'API nous permettant de sélectionner la pizzeria la plus proche et confirmer ou non l'éligibilité du client pour une livraison basée sur un critère de « distance maximale de livraison », Nous souhaitons une réponse sous format Json.

Synthèse:



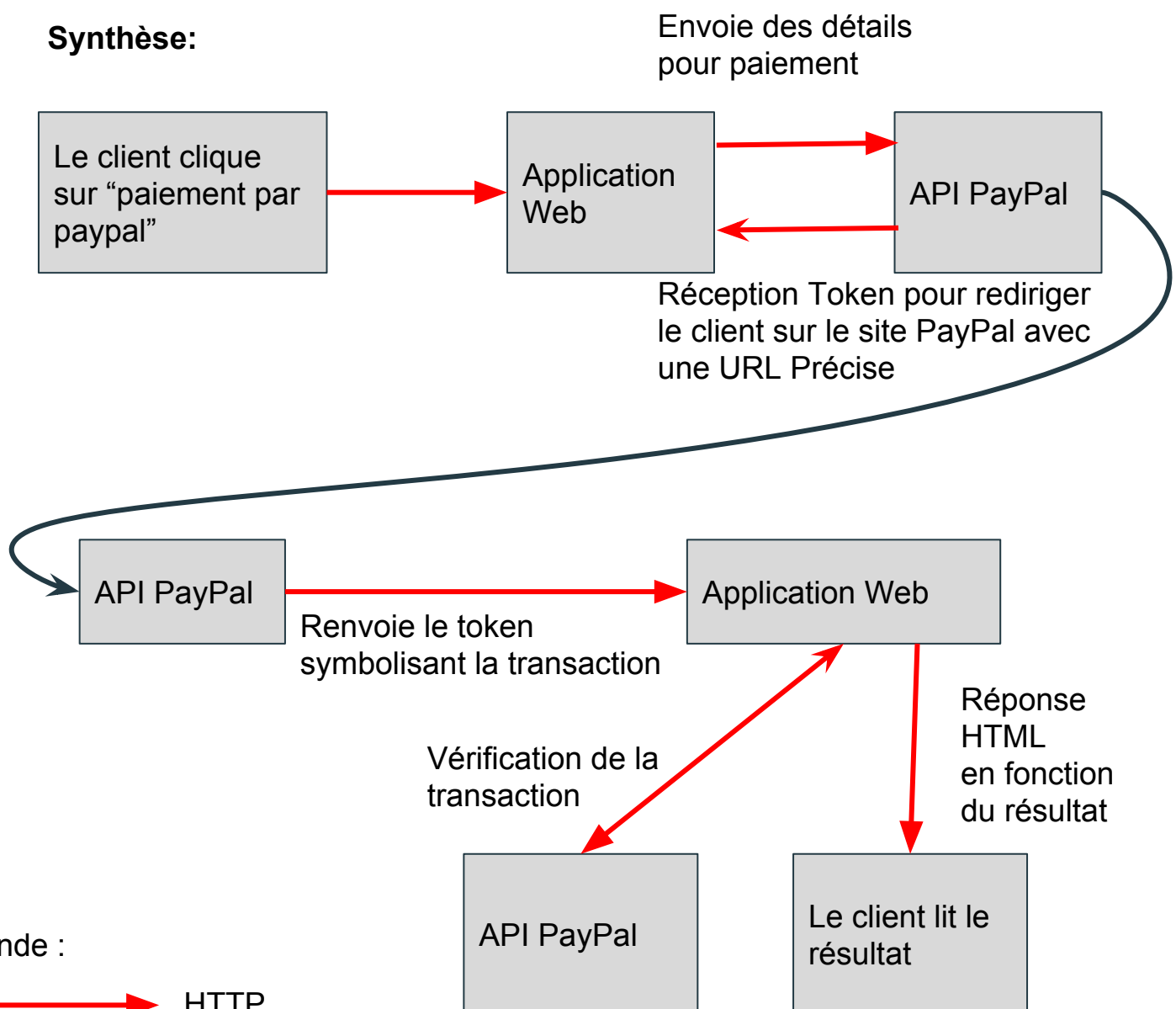
Les composant externes

- Explication d'utilité (suite)

API PayPal

Le système de paiement utilisé est PayPal. Le client clique sur « Paiement avec PayPal », nous faisons appel au service PayPal en lui indiquant dans une requête http les informations de paiement, celui-ci nous réponds en nous envoyant un « token » qu'il nous ait possible d'utiliser pour rediriger le client sur le site PayPal afin d'y finaliser la transaction.

Synthèse:




Déploiement sur le web

- Explication

Hébergement des serveurs :

Nous opterons pour un hébergement sur le cloud ceci permettant notamment de facilement modifier la taille du serveur si le nombre de connexion sur le site internet augmente.

Les serveurs seront hébergés sur  **alwaysdata**
Hébergement simple, rapide et infogéré.