



# OC Pizza

Document de solution technique

IT Consulting & Development

# Choix de l'architecture du système

- L'architecture du projet sera monolithique étant donnée que la structure nécessite peu de ressource et sera amené à peu d'évolution.
- Le développement se fera « From scratch » étant donné la spécificité du projet et que les CMS Python disponible sur le marché ne permettent pas d'intégrer totalement toutes les fonctions nécessaires du projet.
- Néanmoins, pour la mise en oeuvre du projet, le choix d'un framework et des modules intégrables sont choisis pour un réel gain de temps.
- Pour la gestion des données qui ne subirait pas d'analyse poussée, une base de donnée minimaliste fera amplement l'affaire.

# Choix de conception

Liste des solutions techniques principaux retenue dans le développement du système de gestion de la pizzeria.

1.  python

- Langage de programmation interprété et orienté objet

2.  PostgreSQL

- Base de donnée

3. **django**

- Framework open source web basé sur Python

4.  oscar

- Module e-commerce intégrable dans Django

5.  stripe

- Module de paiement en ligne

6. **NGINX**

- Serveur web



**Un framework web puissant et populaire codé sous Python avec une très grande communauté de contributeur.**

**Django est un framework qui s'inspire du principe MTV où la vue est gérée par un gabarit composé de trois parties distinctes :**

- **Un langage de gabarits flexible qui permet de générer du HTML, XML ou tout autre format texte ;**
- **Un contrôleur fourni sous la forme d'un « *remapping* » d'URL à base d'expressions rationnelles ;**
- **Une API d'accès aux données est automatiquement générée par le cadre compatible CRUD. Inutile d'écrire des requêtes SQL associées à des formulaires, elles sont générées automatiquement par l'ORM.**

Source: [Wikipedia](#)

**Il inclus également un serveur web léger permettant de développer et tester ses applications en temps réel sans déploiement.**

**Offre la possibilité d'intégrer différents modules aux besoins de chaque application.**

**Et finalement, le framework fournit également un espace d'administration des données de la base de données, personnalisables et évolutif.**





## Une base de données open source avec plus de 30 ans de développement actif et une grande communauté active

**PostgreSQL est un système de gestion de base de données de classe d'entreprise open source. Il est compatible avec SQL pour les requêtes relationnelles, et avec JSON pour les requêtes non-relationnelles.**

**Les multiples fonctionnalités de PostgreSQL permettent aux développeurs de créer des applications plus facilement, et aident les administrateurs à développer des environnements tolérants aux erreurs en protégeant l'intégrité des données.**

**Ses principales caractéristiques sont une orientation objet et une haute disponibilité. On peut aussi citer sa compatibilité avec de multiples plateformes, langages de programmation et middlewares.**

**Ses fonctionnalités de programmation server-side le démarquent aussi, au même titre que sa prise en charge des architectures réseau client-serveur et sa compliance avec le standard ANSI SQL ainsi que sa compatibilité ANSI-SQL2008. Citons aussi sa fonctionnalité de réplication SSL basée log ou trigger. En outre, la prise en charge de JSON permet le linking vers d'autres Data Stores comme NoSQL.**

Source: [Lebigdata.fr](http://Lebigdata.fr)

**PostgreSQL sera l'idéale pour développer avec Django, qui dispose d'un adaptateur « Psycopg » pour faciliter sa gestion. Sa prise en main simple et une configuration qui n'est pas chronophage sera un atout.**



## Module Django permettant de générer un site e-commerce adaptable et évolutif selon le domaine d'activité.

Oscar est un package-module du framework Django qui est composé de:

- **Un noyau extensible** - toute classe peut être remplacée et étendue. Il s'agit de la fonctionnalité clé qui permet à tout projet d'être personnalisé avec précision dans la base de code.
- **Un ensemble de modèles bien conçu** basé sur l'expérience de nombreux projets de commerce électronique, grands et petits.
- **Documentation complète** comprenant des recettes pour résoudre les problèmes courants.
- **Suite de tests étendue.**

Source: [Oscar commerce](#)

Existant depuis 2013 et compatible avec la version de Python 3, oscar est utilisé dans de diverse project d'e-commerce avec une notation de git considérable.

Source: [Django packages](#)

**Le module permet une implémentation très rapide d'une architecture de commerce en ligne intégrant la gestion de catalogue produit, gestion de la vente, gestion d'utilisateur et suivie comptable. Cette solution est préférable qu'à devoir à développer une architecture**

**Avec une constante évolution et de mise à jour récurrent, ce module sera l'idéale pour être implémenté et personnalisé dans le projet d'OC Pizza.**



## Stripe, service tierce de paiement acceptant une grande catégories de carte bancaire intégrable dans Django

**Stripe est un logiciel tierce qui permet d'intégrer les fonctionnalité destinée au paiement par internet pour les professionnels, recourant à un trafic d'API.**

Source: [Wikipédia](#)

### **Les fonctionnalités principales de Stripe:**

- **Il permet d'accepter en toute sécurité les paiements par carte bancaire en acceptant la plupart des grands types de carte bancaire nationale et internationale.**
- **Le service est sécurisé et respecte les normes les plus strictes en vigueur dans l'industrie et connues dans le monde entier.**
- **Frais de service faible.**

Source: [One.com](#)

**Django-oscar permet l'intégration de Stripe dans sa fonctionnalité de processus de paiement. Le choix éventuel d'une intégration du système de paiement existant reste toutefois possible. Mais, étant donné la taille d'OC Pizza, ce service de paiement en ligne est aisément suffisant pour son fonctionnement.**



## Un serveur Web ainsi qu'un proxy Open Source

**NGINX est conçu pour offrir une faible utilisation de la mémoire et une grande simultanéité. Plutôt que de créer de nouveaux processus pour chaque requête Web, NGINX utilise une approche asynchrone et événementielle où les requêtes sont traitées dans un seul thread.**

**Avec NGINX, un processus maître peut contrôler plusieurs processus de travailleurs. Le maître gère les processus du travailleur, tandis que les travailleurs effectuent le traitement proprement dit. Comme NGINX est asynchrone, chaque requête peut être exécutée simultanément par le travailleur sans bloquer les autres requêtes.**

**Quelques caractéristiques communes vues dans NGINX incluent :**

- **Proxy inversé avec mise en cache**
- **IPv6**
- **Équilibrage de charge**
- **Support FastCGI avec mise en cache**
- **WebSockets**
- **Gestion des fichiers statiques, des fichiers d'index et de l'indexation automatique**
- **TLS/SSL avec SNI**

Source: [Kinsta.com](https://kinsta.com/fr/nginx/)

**Ce serveur web est l'idéale pour gérer une pic trafic entrant à des moments forte demande clientèle. Les réglages sont également rapides. Et fonctionne parfaitement avec le framework Django.**