

# DOSSIER DE CONCEPTION TECHNIQUE

## OC PIZZA



**Auteur**  
Elodie Meunier  
Développeur



- 1. Versions ..... 3
- 2. Introduction..... 4
  - 1. Objet du document ..... 4
  - 2. Références ..... 4
- 3. Architecture technique ..... 5
  - 1. Application web ..... 5
- 4. Architecture de déploiement ..... 6
  - 1. Serveur de base de données ..... 6
  - 2. Serveur ..... 7
- 5. Architecture logicielle ..... 8
  - 1. Principe généraux ..... 8
  - 2. Structure des sources ..... 8
  - 3. Diagramme de classe ..... 9
  - 4. Application web ..... 10
- 6. Points particuliers ..... 11
  - 1. Ressources ..... 11

# 1. VERSIONS

Auteur	Date	Description	Version
Elodie	10/02/2021	Création du document	1

### 2.1 OBJET DU DOCUMENT

---

Le présent document constitue le dossier de conception technique de l'application web [OC Pizza](#) développée par l'entreprise [EdenConception](#). Celui-ci permettra de savoir quels seront les outils utilisés, les méthodes employées et dans quel type d'environnement devra être réalisée l'application.

### 2.2 RÉFÉRENCES

---

Pour de plus amples informations, se référer aux éléments suivants :

1. DCF : Dossier de conception fonctionnelle de l'application
2. DE : Dossier d'exploitation de l'application



## 3.1 APPLICATION WEB

---

Utilisation du framework Django (3.1) développé avec le langage Python (3.8). Celui-ci nous permettra de créer deux interfaces bien distinctes. La première sera consacrée à la partie site web permettant ainsi, à un utilisateur de commander une pizza tandis que la deuxième servira de back office aux différentes équipes d'OC Pizza afin d'assurer le suivi des commandes.

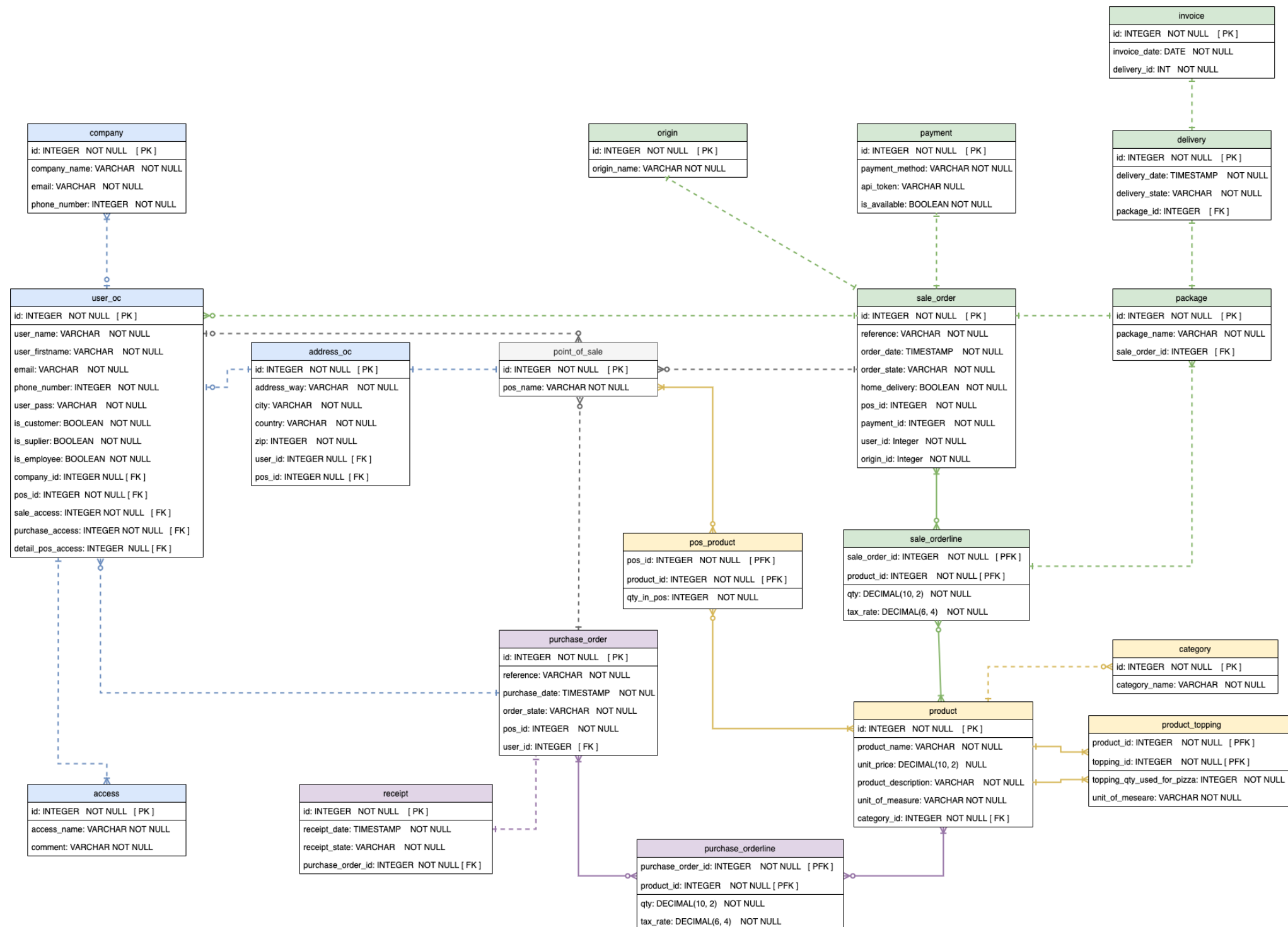




# 4. ARCHITECTURE DE DÉPLOIEMENT

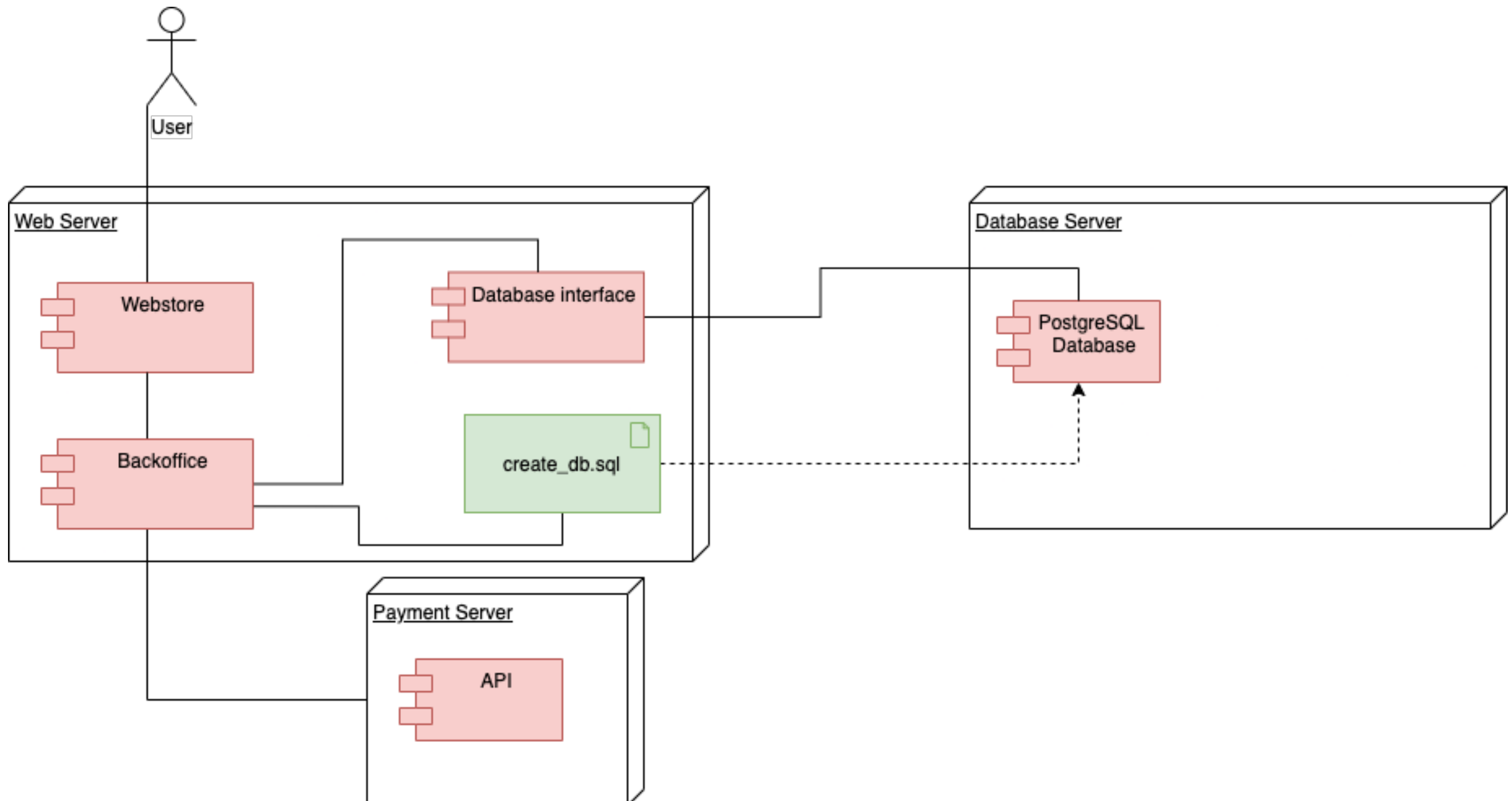
## 4.1 SERVEUR DE BASE DE DONNÉES

Voici le modèle physique de données du futur système. La base de données utilisée sera PostgreSQL (12.5)



## 4.2 SERVEUR

L'application devra être déployée sur un serveur web Gunicorn qui sera relié au serveur de base de données (PostgreSQL 12.5) grâce à un service de déploiement d'application en ligne : Heroku.



## 5.1 PRINCIPE GÉNÉRAUX

Les sources et versions du projet sont gérés par Git, et GitLab. GitLab nous permet également de réaliser des intégrations continue en cas de besoin.

## 5.2 STRUCTURE DES SOURCES

La structuration des répertoires du projet suit la logique de l'architecture **MVT** du framework Django.

(cf. figure 1, figure 2)



```
racine
├── manage.py
├── Procfile
├── README.md
├── requirements.txt
├── ocapp
│   ├── __init__.py
│   ├── settings.py
│   ├── asgi.py
│   ├── urls.py
│   └── wsgi.py
├── contact
│   ├── migrations
│   ├── static
│   │   ├── js
│   │   └── css
│   ├── templates
│   ├── tests
│   ├── models
│   ├── apps.py
│   ├── admin.py
│   ├── urls.py
│   └── views
├── sale
│   ├── migrations
│   ├── static
│   │   ├── js
│   │   └── css
│   ├── templates
│   ├── tests
│   ├── models
│   ├── apps.py
│   ├── admin.py
│   ├── urls.py
│   └── views
```

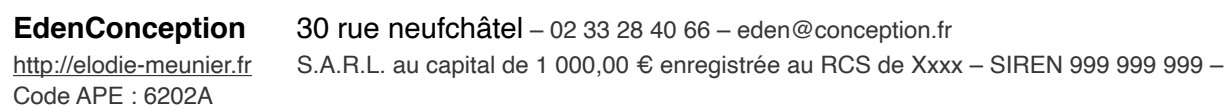
Figure 1

```
purchase
├── migrations
├── static
│   ├── js
│   └── css
├── templates
├── tests
├── models
├── apps.py
├── admin.py
├── urls.py
└── views
product
├── migrations
├── static
│   ├── js
│   └── css
├── templates
├── tests
├── models
├── apps.py
├── admin.py
├── urls.py
└── views
point_of_sale
├── migrations
├── static
│   ├── js
│   └── css
├── templates
├── tests
├── models
├── apps.py
├── admin.py
├── urls.py
└── views
```

Figure 2



Voici le **diagramme de classe** permettant la compréhension de la structure des sources et du projet.



## 5.4 APPLICATION WEB



Voici le diagramme de composants. Ce diagramme est lié au diagramme de déploiement puisqu'il va nous permettre de rentrer dans le détail de celui-ci, en mettant en évidence les dépendances entre les composants.

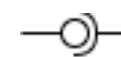
*Voici une légende permettant la lecture de celui-ci.*



**Composant** (permettant de fournir et d'utiliser un comportement).



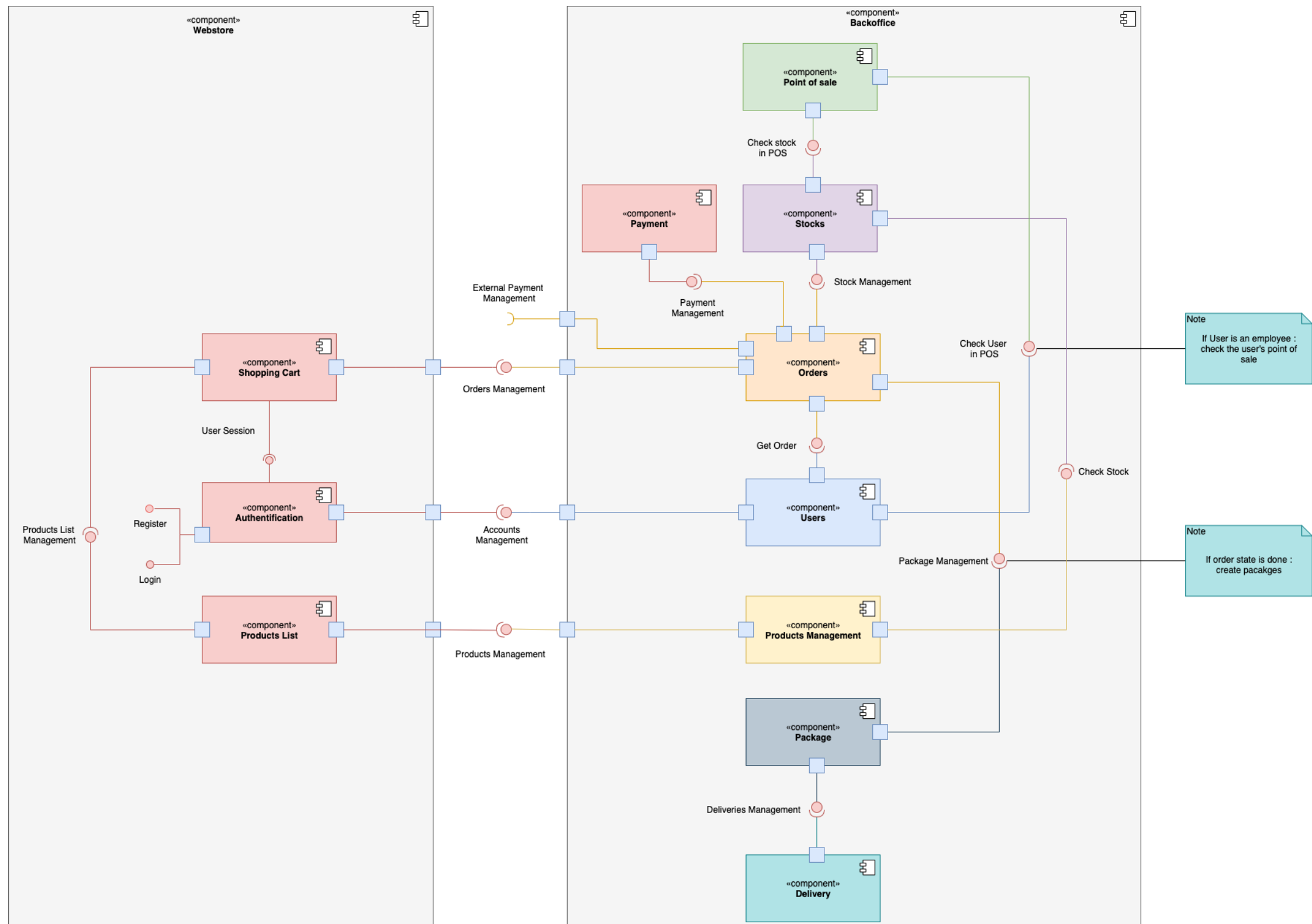
**Port** (spécifie un point d'interaction entre le composant et son environnement).



Le cercle plein indique qu'il s'agit d'une **interface fournie** tandis que le demi-cercle requiert une interface (il s'agit donc d'une **interface requise**).



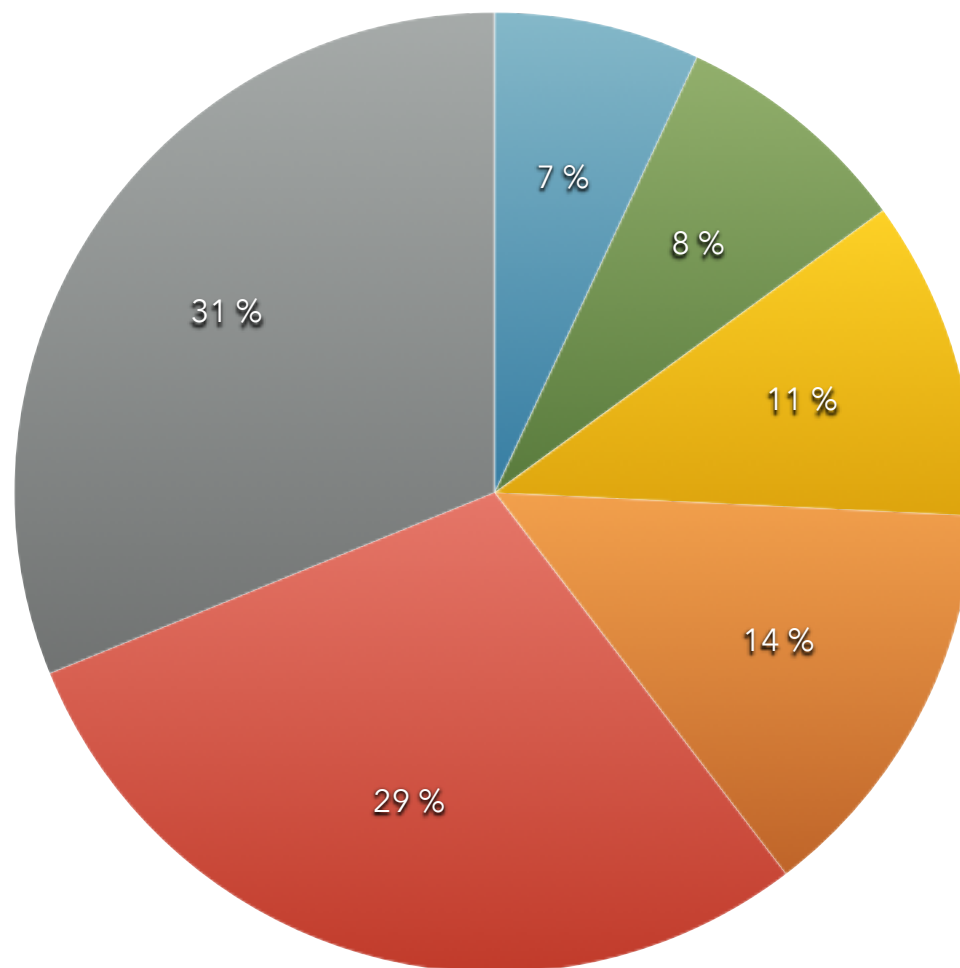
## 5.4 APPLICATION WEB



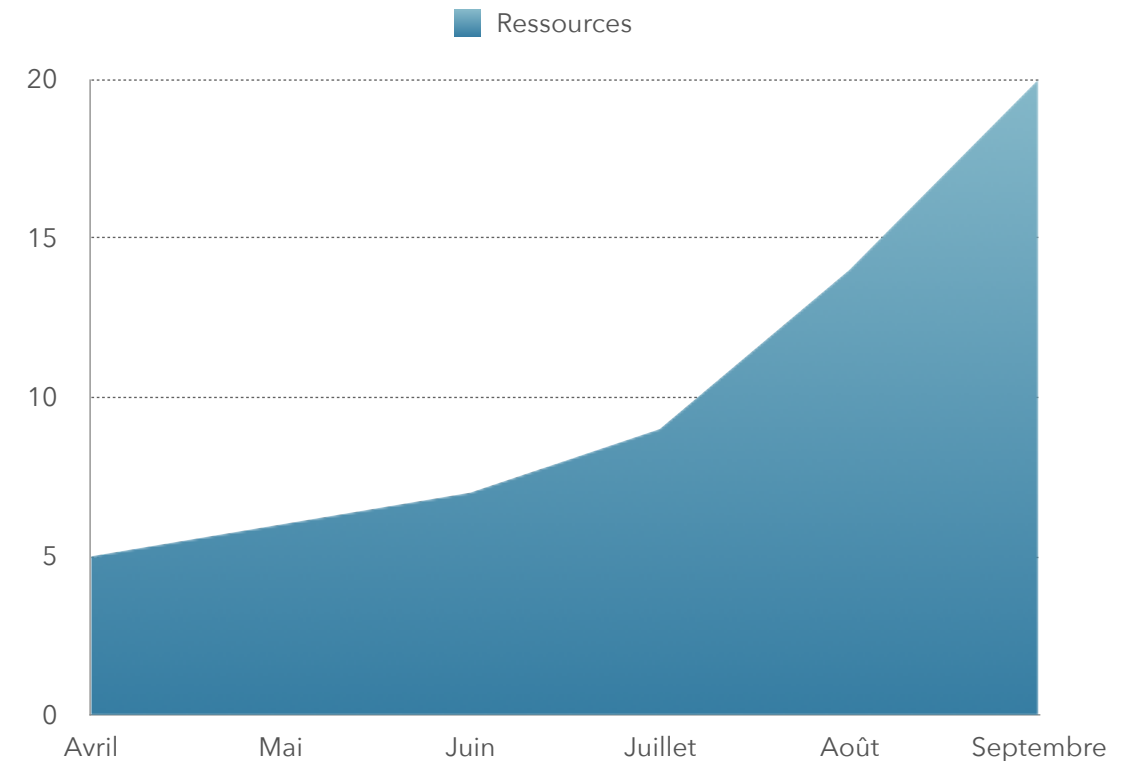
## 6.1 RESSOURCES

L'entreprise EdenConception a réalisé un rétro-planning afin de déterminer le nombre de ressources disponibles pour la réalisation du projet et ainsi spécifier la durée totale du projet.

● Avril ● Mai ● Juin ● Juillet ● Août ● Septembre



Graph 1 - Avancement projet



Graph 2 - Nb de ressources disponible

Sur le [graph 1](#), le pourcentage représente l'avancée du projet en fonction du nombre de ressources disponibles pour ce mois (*cf. graph 2*).

Nous estimons donc que l'entreprise EdenConception aura besoin de 6 mois pour réaliser le projet de l'entreprise OC Pizza.



FIN

