|  |
| --- |
| {{Client}}  {{Projet}}  Dossier de conception technique  Version {{Version}} |
| **Auteur**  {{Auteur}}  {{Auteur\_Role}} |

Table des matières

1 -Versions 3

2 -Introduction 4

2.1 -Objet du document 4

2.2 -Références 4

3 -Architecture Technique 5

3.1 -Composants généraux 5

3.1.1 -Package A 5

3.1.1.1 -Composant X 5

3.1.1.2 -Composant Y 5

3.1.2 -Package B 5

3.1.2.1 -Composant Z 5

3.2 -Application Web 5

3.2.1 -Composants X 5

3.2.2 -Composants Y et Z 5

3.3 -Application XXX... 5

4 -Architecture de Déploiement 6

4.1 -Serveur de Base de données 6

4.2 -Serveur XXX 6

5 -Architecture logicielle 7

5.1 -Principes généraux 7

5.1.1 -Les couches 7

5.1.2 -Les modules 7

5.1.3 -Structure des sources 7

5.2 -Application Web 8

5.3 -Application Xxx 8

6 -Points particuliers 9

6.1 -Gestion des logs 9

6.2 -Fichiers de configuration 9

6.2.1 -Application web 9

6.2.1.1 -Datasources 9

6.2.1.2 -Fichier xxx.yyy 9

6.2.2 -Application Xxx 9

6.3 -Ressources 9

6.4 -Environnement de développement 9

6.5 -Procédure de packaging / livraison 9

6.6 -XXX 9

7 -Glossaire 10

# Versions

|  |  |  |  |
| --- | --- | --- | --- |
| Auteur | Date | Description | Version |
| Gilles DAVID | 21/11/2021 | Création du document | 1 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Objet du document

Le présent document constitue le dossier de conception technique de l'application « OC Pizza »

Objectif du document, est de définir ce qui est nécessaire à la conception de la solution technique proposée.

Les éléments du présents dossiers découlent :

* du dossier de conception technique de l'application
* du dossier de conception fonctionnelle

## Références

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

1. **DCT** : pathFile Dossier de conception technique de l'application
2. **DCF** : Dossier de conception fonctionnelle

# Architecture Technique

## Composants généraux

### Package « Back-Office »

#### Composant NodeJS

**NodeJS** (Node Package Manager – version 7.12)

Permet d’exécuter du code « **JavaScript** », nécessaire pour exécuter « **Express** » et « **ReactJS** »

#### Composant Express

**Express** (version 4.17)

Framework pour NodeJS qui permet de gérer le serveur, reçoit les appel réseaux, interroge la base données et renvoie les données demandées avec le niveau de sécurité préalablement défini

### Package IOS

#### Composant application IOS

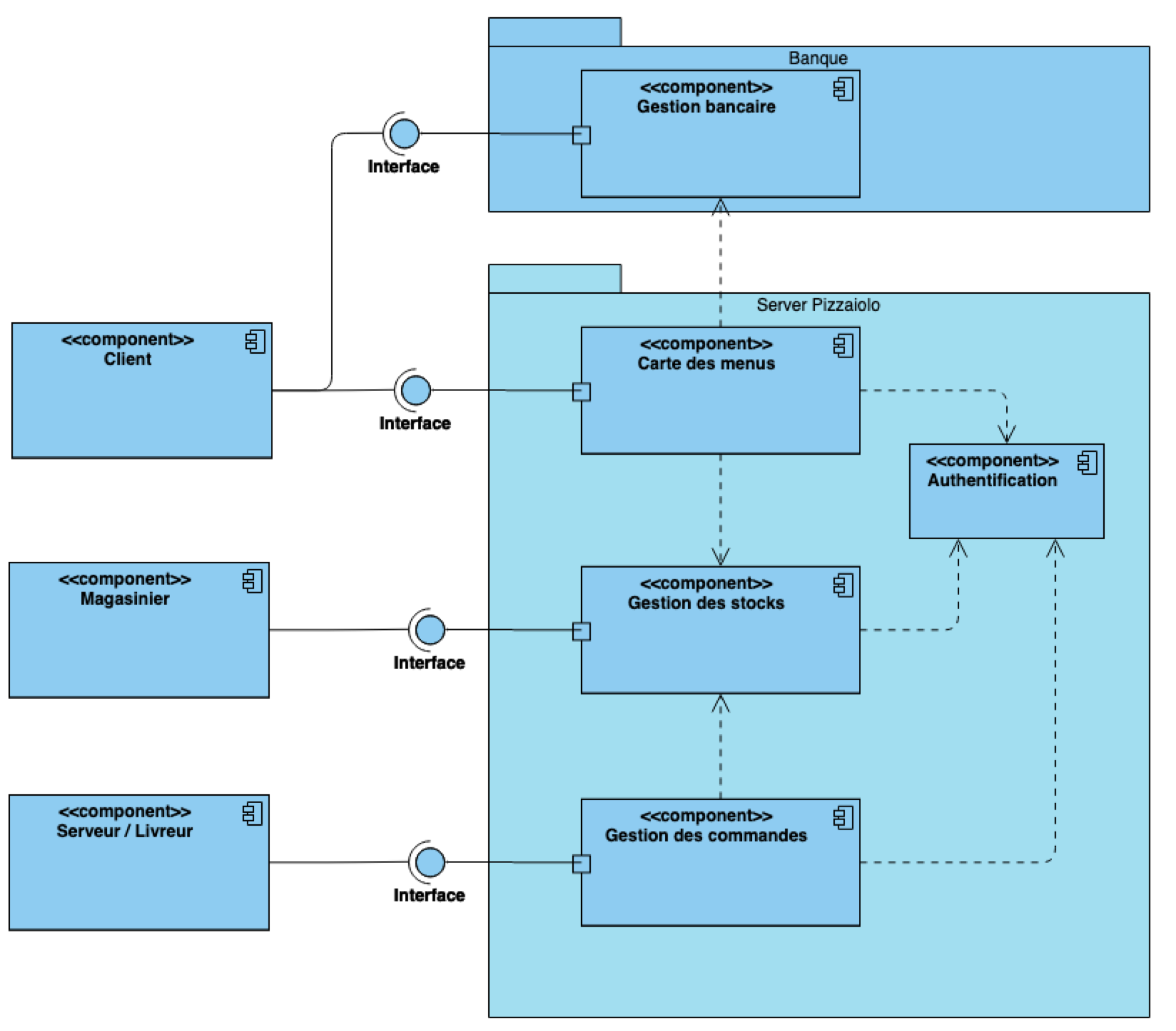
Application sur smartphone qui permet d’interroger le serveur basé sur Express, pour réaliser les opérations du type commande sur le serveur «  OC Pizza »

## Application Web

La pile logicielle est la suivante :

* **NodeJS** (Node Package Manager – version 7.12)
* **ReactJS** (version 17)

Diagramme UML de Composants



### Composants « client », « Magasinier », et Serveur/Livreur »

Les composants « client », « Magasinier », et Serveur/Livreur » représente les personnes qui vont pouvoir agir sur le système via une interface web ou de type application IOS

### Composants « Carte des menus »

Représente la partie logiciel qui traite des menus, reçoit les appels venant du client, renvoi la partie demandée par le client.

### Composants « gestion des stocks »

Représente la partie logiciel qui va permettre au magasinier de mettre à jour les valeurs correspondant à l’état des stocks, dans la base de données

### Composants « gestion des commandes »

Représente la partie logiciel qui permet de consulter l’état d’une commande, défini dans la base, au moment présent de la consultation.

### Composants « authentification »

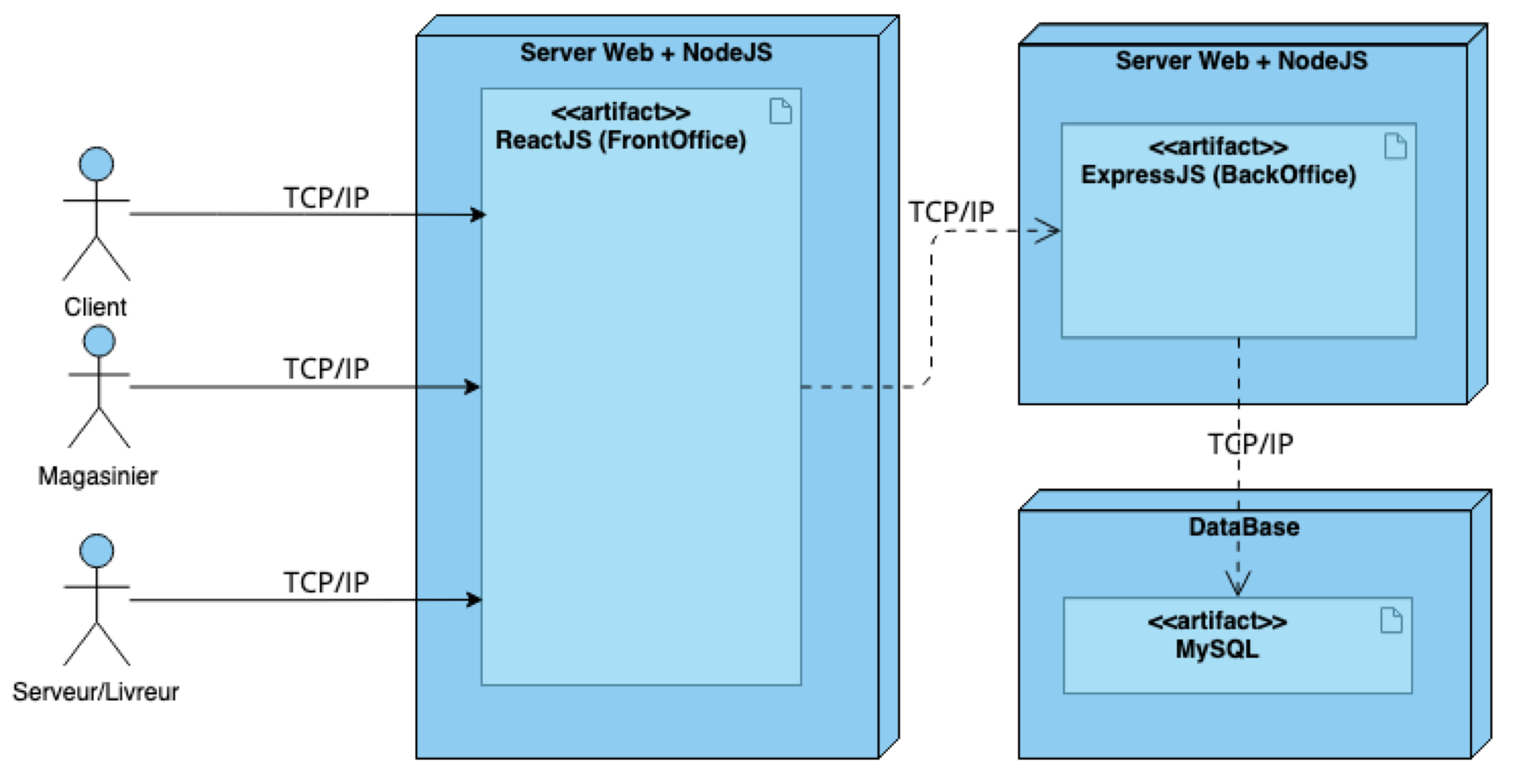
Représente la partie logiciel qui gère les « login » et « mot de passe » afin de garantir la reconnaissance d’une personne ou d’un type d’employé (serveur/magasinier/livreur/patron), d’ajouter de nouvelle personne, ainsi que de les modifier.

## Application smartphone

L’application smartphone permet de se connecter au serveur, de consulter la carte des menus, de modifier l’état des stocks

# Architecture de Déploiement

Diagramme UML de déploiement



## Serveur de Base de données

Serveur OVHCloud ( <https://www.ovhcloud.com/fr/web-hosting/options/start-sql/> ) avec 800Mo, Jusqu'à 30 connexions simultanées et back-up inclus. "MySQL" version 5.7

## Serveur « OVH »

Serveur OVH ( <https://www.ovhcloud.com/fr/bare-metal/rise/prices/> à 61,74 € HT/mois) :

Processeur : Intel Xeon-E 2136 - 6 c / 12 t - 3.3 GHz / 4.5 GHz   
Mémoire : À partir de 32Go   
Stockage : NVMe, SATA disponible

Bande passante publique : 500 Mbit/s

# Architecture logicielle

## Principes généraux

Les sources et versions du projet sont gérées par **Git**, les dépendances et le packaging par **NPM**

### Les couches

L'architecture applicative est la suivante :

* unecouche **business** : responsable de la logique métier du composant
* unecouche **model** : implémentation du modèle des objets métiers
* …
  + …
* ...

### Les modules

Ex: modules Maven dans le cas d’application multi-module...

### Structure des sources

La structuration des répertoires du projet suit la logique suivante :

* les répertoires sources sont créés de façon à respecter la philosophie Maven (à savoir : « convention plutôt que configuration »)

racine  
 ├─ *pom.xml*  
 ├─ <moduleX>  
 │ ├─ *pom.xml*  
 │ └─ src  
 │ ├─ main  
 │ │ ├─ java  
 │ │ └─ resources  
 │ └─ test  
 │ ├─ java  
 │ └─ resources  
 └─ src  
 └─ lib

* ...

## Application Web

Si besoin, diagramme UML de composants pour monter les différents modules et leur inter-dépendances

## Application smartphone

# Points particuliers

## Gestion des logs

…

## Fichiers de configuration

### Application web

...

#### Datasources

...

#### Fichier xxx.yyy

...

### Application Xxx

...

## Ressources

...

## Environnement de développement

## Procédure de packaging / livraison

## XXX

…

# Glossaire

|  |  |
| --- | --- |
|  |  |
|  |  |