

Oc Pizza

Solution Informatique Complète

Dossier de conception technique

Version 1.0

Auteur
Benoit Lefèvre
Analyste Développeur

TABLE DES MATIERES

1 - VERSIONS	4
2 - INTRODUCTION	5
2.1 - OBJET DU DOCUMENT	5
2.2 - RÉFÉRENCES	5
3 - ARCHITECTURE TECHNIQUE	6
3.1 - COMPOSANTS GÉNÉRAUX	6
3.2 - DESCRIPTIF DES COMPOSANTS	6
3.2.1 - <i>Les composants extérieurs au système</i>	6
3.2.1.1 - Service Bancaire	6
3.2.1.2 - Applicatif Fournisseur	6
3.2.2 - <i>Les composants internes au système</i>	7
3.2.2.1 - Applicatif OcPizzaRestaurant	7
3.2.2.2 - Base de données PostgreSQL 12	7
3.2.2.2.1 Base de données locale	7
3.2.2.2.2 Base de données globale	7
3.2.2.3 - Application web OcPizzaWeb	8
3.2.2.4 - Application web OcPizzaWebManager	8
4 - ARCHITECTURE DE DÉPLOIEMENT	9
4.1 - SERVEUR DE BASE DE DONNÉES LOCALE	9
4.1.1 - <i>Matériel</i>	9
4.1.2 - <i>Caractéristiques techniques</i>	10
4.2 - SERVEUR DE BASE DE DONNÉES GLOBALE	10
4.2.1 - <i>Caractéristiques techniques</i>	10
4.3 - SERVEUR DÉDIÉ HÉBERGÉ	10
4.3.1 - <i>Matériel</i>	10
4.3.2 - <i>Caractéristiques techniques</i>	10
4.4 - PÉRIPHÉRIQUES ÉTABLISSEMENT	11
5 - ARCHITECTURE LOGICIELLE	12
5.1 - PRINCIPES GÉNÉRAUX	12
5.1.1 - <i>Les couches</i>	12
5.1.2 - <i>Structure des sources</i>	12
6 - MODÈLE PHYSIQUE DE DONNÉES	14
7 - POINTS PARTICULIERS	15
7.1 - GESTION DES LOGS	15
7.2 - FICHIERS DE CONFIGURATION	15
7.2.1 - <i>Application web</i>	15
7.2.2 - <i>PostgreSQL</i>	15
7.2.3 - <i>Apache Tomcat</i>	15
7.3 - RESSOURCES	15
7.4 - ENVIRONNEMENT DE DÉVELOPPEMENT	16
7.5 - PROCÉDURE DE PACKAGING / LIVRAISON	16

8 - GLOSSAIRE	17
----------------------------	-----------

1 - VERSIONS

Auteur	Date	Description	Version
Benoit Lefèvre	15/10/2019	Création du document	1.0

2 - INTRODUCTION

2.1 - Objet du document

Le présent document constitue le dossier de conception technique de la solution informatique proposée.

Ce document a pour objectif de décrire le plus fidèlement et le plus clairement possible :

- Les composants propres à la solution ainsi que les composants extérieurs et l'ensemble des interactions
- Le déploiement de la solution proposée
- L'architecture logicielle
- Le Modèle Physique de Données
- Les points particuliers

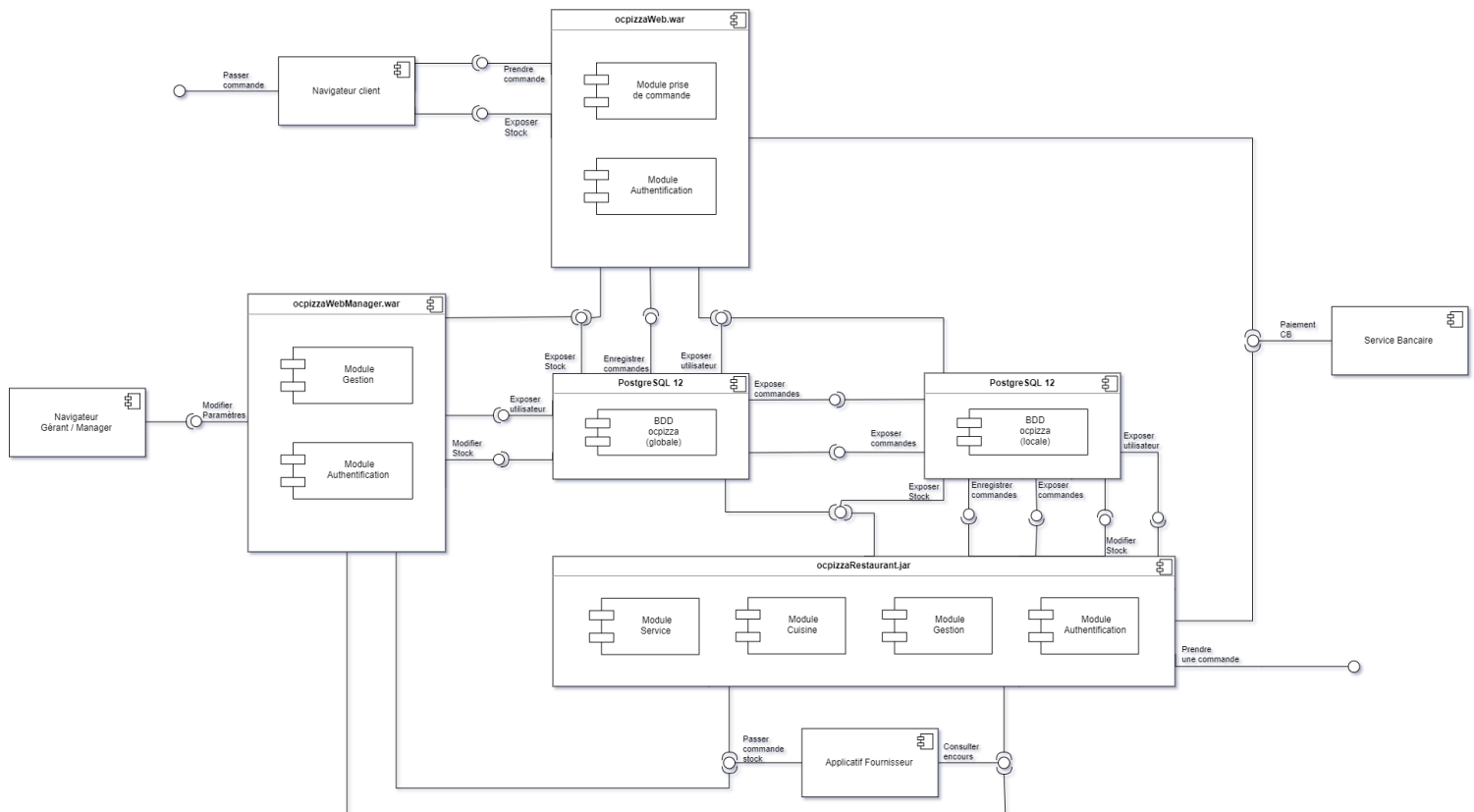
2.2 - Références

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

1. **DCF - OcPizza** : Dossier de conception fonctionnelle de l'application
2. **DE – OcPizza** : Dossier d'exploitation de la solution informatique

3 - ARCHITECTURE TECHNIQUE

3.1 - Composants généraux



3.2 - Descriptif des composants

3.2.1 - Les composants extérieurs au système

3.2.1.1 - Service Bancaire

Le composant Service Bancaire représente un composant extérieur au système permettant d'accepter les paiements par carte bancaire. Différentes solutions existent comme des fournisseurs tels que PayPal et Stripe mais vous pouvez opter pour une solution proposée par une banque traditionnelle comme par exemple E-transaction du Crédit Agricole. Toutes ces solutions ont comme point commun de fournir un module simple à intégrer à une application web ou à un applicatif.

3.2.1.2 - Applicatif Fournisseur

Le composant Applicatif Fournisseur représente un composant extérieur au système permettant de passer

une commande de produits bruts au sein d'un catalogue fournisseur.

Différentes solutions existent comme des logiciels tiers tel que SAP où différents fournisseurs peuvent donner accès à leurs catalogues, des sites web ou des API permettant d'intégrer un système de commande à un progiciel.

3.2.2 - Les composants internes au système

3.2.2.1 - Applicatif OcPizzaRestaurant

Le composant applicatif OcPizzaRestaurant représente l'ensemble des modules déployés au sein d'un établissement. Il permet de mettre en évidence les relations nécessaires avec les composants extérieurs et avec la base de données locale.

De par l'architecture proposée, ce composant sera totalement évolutif voire même remplaçable à condition d'offrir les mêmes interfaces offertes et requises et cela sans avoir à transformer l'intégralité de l'infrastructure.

La pile logicielle est la suivante :

- Application **JAVA** (JDK version 1.8)

3.2.2.2 - Base de données PostgreSQL 12

Il y a deux composants distincts qui sont des bases de données PostgreSQL :

- La base de données locale propre à chaque établissement
- La base de données globale centralisant l'ensemble des informations contenues par chaque base de données locales ainsi que les commandes internet.

3.2.2.2.1 Base de données locale

Nous vous avons proposé dans notre solution de déployer un serveur avec base de données sur chacun de vos établissements afin de vous garantir une continuité de service, peu importe les problèmes de connexion internet qui pourraient survenir.

Comme on peut le constater, la base de données locale présente de nombreuses interactions avec l'applicatif OcPizzaRestaurant mais aussi avec la base de données centrale.

Toutes les données propres à un établissement sont stockées sur la base de données de l'établissement et celle-ci récupère aussi les commandes passées par internet sur la base de données centrale.

3.2.2.2.2 Base de données globale

Le composant Base de données globale permet de centraliser l'ensemble des données de chaque établissement et de stocker les commandes passées par internet via l'application web OcPizzaWeb.

On observe donc que la communication entre les bases de données centrale et locale se font dans les deux sens afin de récupérer en permanence les données nécessaires à l'exploitation de vos établissements.

La base de données globale interagit aussi avec l'application web OcPizzaWebManager afin de permettre de modifier aux gérants et manager un paramètre propre à n'importe quel établissement.

3.2.2.3 - Application web OcPizzaWeb

L'application web OcPizzaWeb représente toute la logique implantée sur un serveur d'application afin de permettre à un client par exemple de passer une commande en fonction des produits disponibles, de gérer son compte client ou encore de consulter ses commandes en cours ou terminées.

Ce composant communique avec la base de données centrale afin d'y enregistrer les commandes passées par les clients mais aussi pour connaître en temps réel les stocks de chaque établissement ou récupérer toutes informations utiles et nécessaires quant au compte client.

La pile logicielle est la suivante :

- Application **J2EE** (JDK version 1.8)
- Serveur d'application **Tomcat 9.0**

3.2.2.4 - Application web OcPizzaWebManager

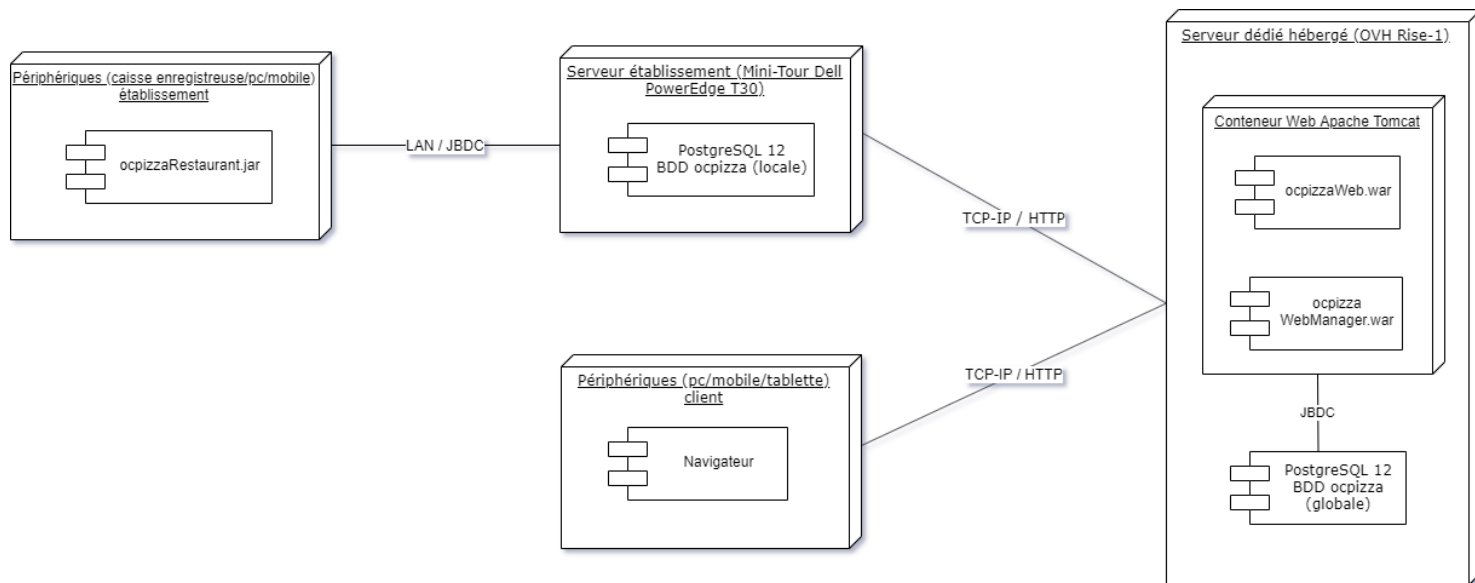
L'application OcPizzaWebManager représente toute la logique implémenter sur un serveur d'application afin de permettre à un gérant ou à un manager d'interagir avec le module gestion afin de pouvoir ajouter, modifier ou supprimer un élément au sein du système.

Ce composant communique avec la base de données globale afin d'y enregistrer toutes modifications nécessaires (répercutée automatiquement auprès de l'établissement concerné). Ce composant communique aussi avec les applicatifs fournisseurs.

La pile logicielle est la suivante :

- Application **J2EE** (JDK version 1.8)
- Serveur d'application **Tomcat 9.0**

4 - ARCHITECTURE DE DEPLOIEMENT



Comme décrit dans la documentation fonctionnelle de la solution informatique proposée, la solution est composée de deux interfaces distinctes :

- Une interface client matérialisée par l'application web OcPizzaWeb
- Une interface établissement matérialisée à la fois par l'appliquatif OcPizzaRestaurant et par l'application web OcPizzaWebManager

4.1 - Serveur de Base de données locale

Nous vous avons proposé dans notre solution de déployer un serveur hébergeant une base de données sur chacun de vos établissements afin de vous garantir une continuité de service, peu importe les problèmes de connexion internet qui pourraient survenir.

Toutes les données propres à un établissement sont stockées sur la base de données de l'établissement et celle-ci récupère aussi les commandes passées par internet sur la base de données centrale.

Ce serveur de base de données hébergera le schéma public de la base OcPizza et communiquera avec l'appliquatif OcPizzaRestaurant ainsi qu'avec le serveur de base de données globale.

4.1.1 - Matériel

Serveur mini-tour Dell PowerEdge T30 :

- Processeur Intel® Xeon® E3- 1225 v5
- 8Go DDR4 2133MHz
- 1TB 7.2K RPM SATA 6Gbps

- Intel I219-LM 1 port Gigabit Ethernet LAN 10/100/1000

4.1.2 - Caractéristiques techniques

- OS : Windows Server 2016
- SGBD : PostgreSQL 12

4.2 - Serveur de base de données globale

La base de données globale sera déployée sur le serveur dédié où sera aussi installé le serveur d'application Apache Tomcat. Pour connaître le matériel concerné par ce déploiement, merci de vous rendre à la section [4.3.2 Caractéristiques techniques](#).

Ce serveur de base de données hébergera le schéma public de la base OcPizza et communiquera à la fois avec le serveur d'application mais aussi avec l'ensemble des serveurs de base de données déployés sur chacun de vos établissements.

4.2.1 - Caractéristiques techniques

- OS : Windows Server 2016
- SGBD : PostgreSQL 12

4.3 - Serveur dédié hébergé

Le déploiement des applications web nécessitent un serveur dédié capable de prendre en charge un fort trafic, c'est pourquoi un serveur dédié sera loué auprès d'OVH.

4.3.1 - Matériel

Serveur physique hébergé Rise-1 :

- Intel Xeon E3-1230v6 - 4c/ 8t - 3.5GHz/ 3.9GHz
- 32 Go DDR4 ECC 2133MHz
- 2x450Go SSD NVME Soft RAID

4.3.2 - Caractéristiques techniques

- OS : Windows Server 2016
- Serveur d'application : Apache Tomcat 9.0.26
- SGBD : PostgreSQL 12

4.4 - Périphériques établissement

L'appliquatif OcPizzaRestaurant sera déployé sur l'ensemble des périphériques de vos établissement :

- Caisses enregistreuses
- Ordinateurs + écran déporté en cuisine
- Tablettes

L'ensemble de ses périphériques et le serveur local seront reliés entre eux par un réseau Lan leur permettant de communiquer.

5 - ARCHITECTURE LOGICIELLE

5.1 - Principes généraux

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

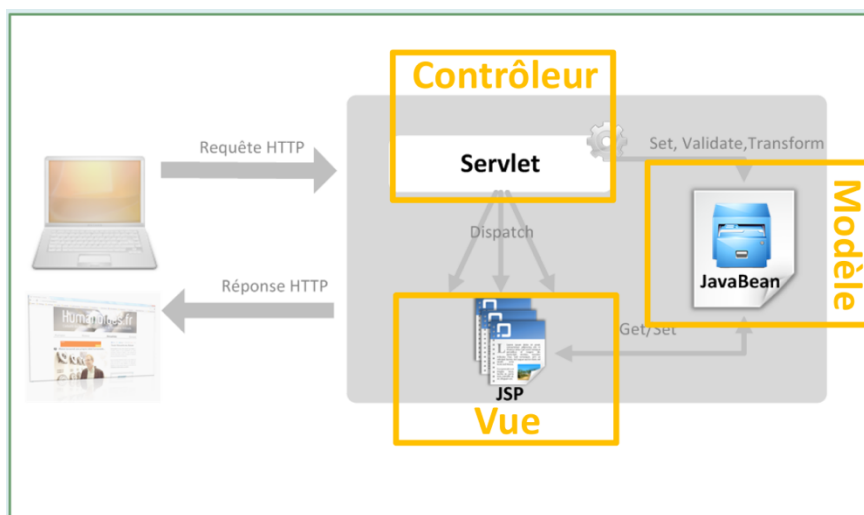
Gradle simplifie le packaging de projet grâce à l'application de plugin. Pour les différents composants de la solution, nous avons utilisé les plugins :

- Java plugin pour la production des .jar
- War plugin pour la production des .war

5.1.1 - Les couches

L'architecture applicative est une architecture MVC :

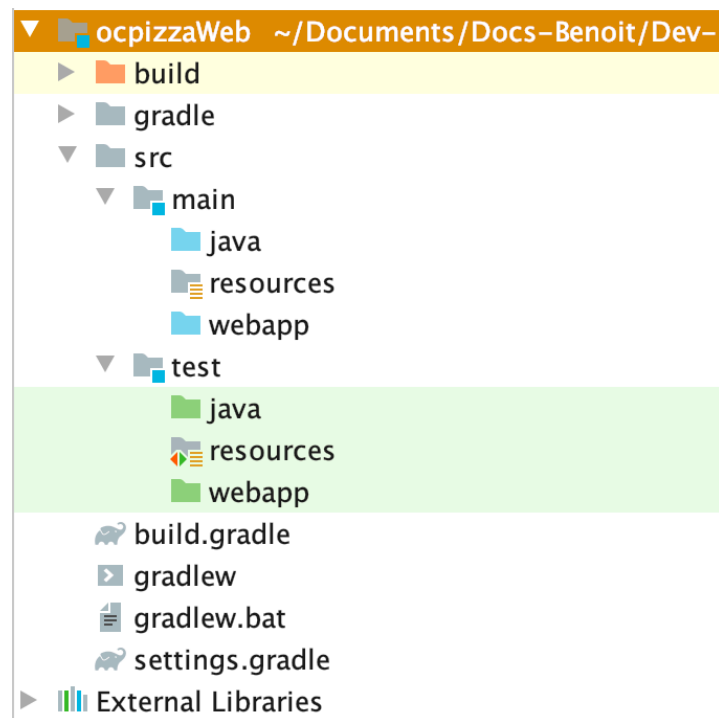
- une couche **Model** : responsable de la logique métier du composant et de la manipulation des données en BDD
- une couche **View** : responsable de l'interface perçue par l'utilisateur
- une couche **Controller** : responsable de la gestion des actions sur la vue



5.1.2 - Structure des sources

La structuration des répertoires du projet suit la logique suivante :

- Les répertoires sources sont créés de façon à respecter la philosophie Gradle (à savoir : « convention plutôt que configuration »)



6 - MODELE PHYSIQUE DE DONNEES

Le Modèle Physique de Données est issu du diagramme de classe UML présenté dans le DCF – OcPizza à la section 4 - Domaine Fonctionnel.

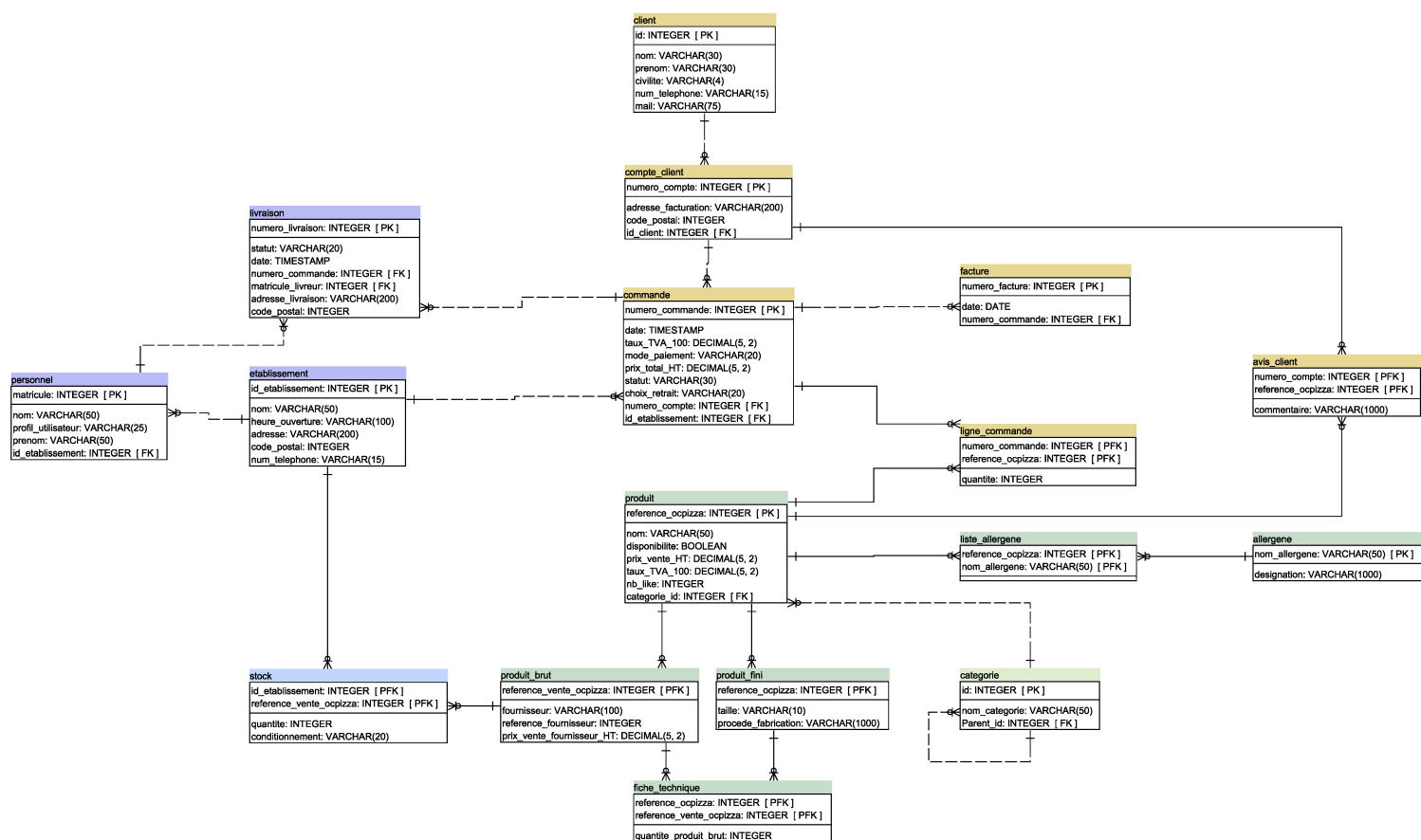
L'ensemble des classes présentes dans le diagramme de classe se retrouvent présentes dans le Modèle Physique de Données sous la forme de table.

Chaque table possède une clé primaire qui peut être propre à la table [PK] ou issu d'une autre table [FK]. La clé primaire d'une table peut être composée de plusieurs clés, on parle alors de clé primaire composite.

Au sein d'une table, il est possible de faire référence à la clé primaire d'une autre table pour matérialiser certaines relations entre 2 tables grâce à une clé étrangère [FK].

Une table « liste_allergène » est présente dans le Modèle Physique de Données afin de faire la relation entre la table « produit » et la table « allergène » qui est une relation « plusieurs à plusieurs ». Cette table n'est pas présente dans le diagramme de classe du domaine fonctionnel puisque qu'elle ne caractérise pas la relation en question.

Le schéma public de la base de données Oc Pizza est directement issu de ce modèle physique de données.



7 - POINTS PARTICULIERS

7.1 - Gestion des logs

Par défaut, nous avons configuré les fichiers de configuration de log sur un niveau de criticité WARN. Seules les informations supérieures ou égales à ce niveau de criticité seront loggées.

Si vous souhaitez modifier le niveau de criticité des logs des différents composants de la solution, vous pourrez modifier les fichiers suivants :

Apache Tomcat : logging.properties

PostgreSQL : postgresql.conf

OcPizzaWeb : logging.properties

OcPizzaWebManager : logging.properties

OcPizzaRestaurant : logging.properties

7.2 - Fichiers de configuration

La description de l'ensemble des fichiers de configurations cités ci-dessous est décrite dans le Dossier d'exploitation de la solution aux sections évoquées.

7.2.1 - Application web

web.xml (section 4.3.2 du DE)

7.2.2 - PostgreSQL

pg_hba.conf (section 4.2.2.1 du DE)

pg_ident.conf (section 4.2.2.2 du DE)

postgresql.conf (section 4.2.2.3 du DE)

7.2.3 - Apache Tomcat

context.xml (section 4.1.3.1 du DE)

server.xml (section 4.1.3.2 du DE)

tomcat-users.xml (section 4.1.3.3 du DE)

web.xml (section 4.1.3.4 du DE)

7.3 - Ressources

La plupart des composants du système nécessite l'installation de la ressource suivante :

- JDK ou JRE java (dernière version JAVA SE 13)
<https://www.oracle.com/technetwork/java/javase/downloads/jdk13-downloads-5672538.html>

7.4 - Environnement de développement

L'environnement de développement utilisé est le suivant :

- OS : Windows
- IDE : Eclipse for Java EE
- TaskRunner : Gradle 5.6.2
- JDK : 1.8
- Versionning : Git
- Serveur de dev / pré-prod : Serveur mini-tour Dell PowerEdge T30 sous Windows server 2016
- Serveur d'application : Apache Tomcat 9.0
- SGBD : PostgreSQL 12

7.5 - Procédure de packaging / livraison

Le packaging des différents composants du système a été réalisé grâce à Gradle. En fonction du plugin utilisé et renseigné dans le fichier build.gradle, la tâche build fournie par Gradle nous permet de packager un .war avec le plugin 'war' et un .jar avec le plugin 'java'.

Lors de l'appel à la tâche build, Gradle exécute un ensemble de tâche qui sont les suivantes :

- Task :compileJava
- Task :processResources
- Task :classes
- Task :jar (ou war)
- Task :assemble
- Task :compileTestJava
- Task :processTestResources
- Task :testClasses
- Task :test
- Task :check
- Task :build

8 - GLOSSAIRE
