|  |
| --- |
| Alegia  **Site Alegia**  Version 1.0 |
| **Auteur**  R.Yoann  Développeur |

**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** |
| R.Yoann | 17/02/2020 | Création du document | 1.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

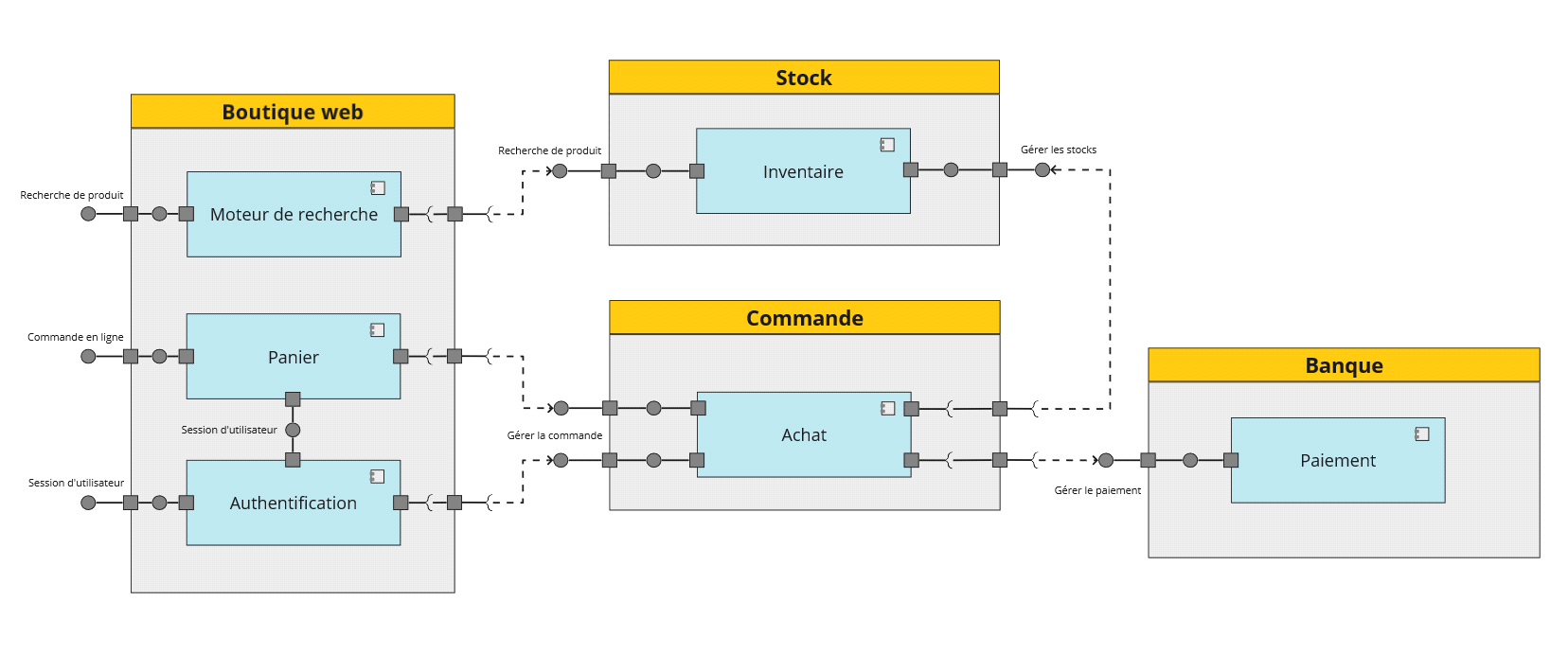
* Introduction
* Objet du document

Le présent document constitue le dossier de conception technique de l'application pizzeria

Objectif du document est de décrire l'architecture du logiciel en passant par des diagrammes de déploiement et composant.

Les éléments du présent dossier découlent :

* Un diagramme de composant
* Un diagramme de déploiement
* Les couches de l'application
* Architecture Technique
* Composants généraux



* *Package "Boutique web"*
* ***Composant "Panier"***

Gère la commande du client, le client peut passer une commande avec plusieurs articles.

* ***Composant "Authentification"***

Gère le compte client. Le client doit posséder un compte pour remplir son panier et effectuer une commande

* *Package "Commande"*
* ***Composant "Achat"***

Gère l'achat du client. Demande confirmation auprès de la banque du paiement.

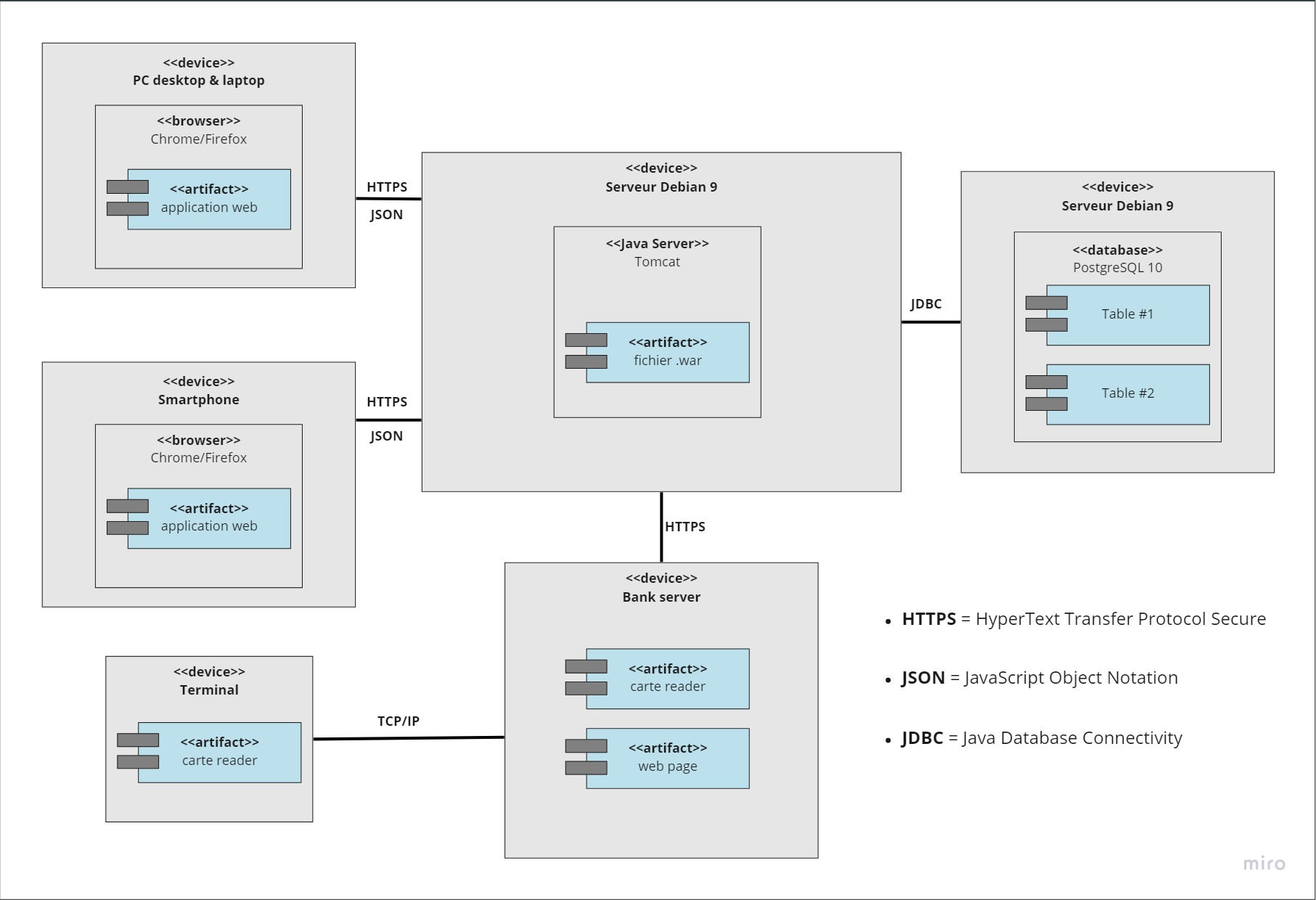
* *Package "Banque"*
* ***Composant "Paiement"***

Gère le paiement du client pour sa commande. Informe la commande sur le statut de la transaction.

* Application Web

La pile logicielle est la suivante :

* Application **J2EE** (JDK version 1.8)
* Serveur d'application **Tomcat 9.0.19**
* Architecture de Déploiement



L'application sera déployée sur un serveur Debian 9, la base de données fonctionnera aussi sur celui-ci.

Le serveur Java est déployé via Tomcat.

La base de données fonctionne avec PostgreSQL 10.

La communication avec l'application web se fait en HTTPS pour sécuriser les clients.

La partie front fonctionne avec Struts 2 et récupère via Feign des informations en JSON venant des microservices.

Pour le paiement, l'application est mise en communication avec une page de la banque en HTTPS.

* Serveur

**Debian 9**

**Caractéristiques techniques :**

Cpu Intel Xeon

1vCore 3.7Ghtz

2Go de RAM

2Go Vswap SSD (450 Mo/s)

SSD 20 Go RAID 10

100Mbs

* Architecture logicielle
* Principes généraux

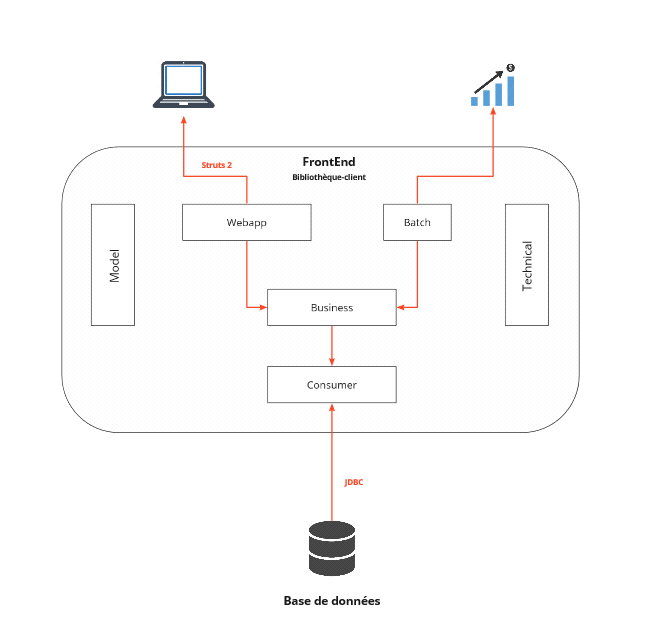
Les sources et versions du projet sont gérées par **Git**, les dépendances et le packaging par **Apache Maven**, **Java/javascript** comme langages utilisées. Framework : **Spring/JQuery/Struts 2**

* *Les couches*

L'architecture pour la partie Back de l’application est la suivante :

* Proxies : des interfaces qui fonctionne avec Feign pour récupérer les données des microservices.
* Action : contient les classes Action qui communique avec les pages JSP.
* Interceptor : gère l’authentification des utilisateurs.
* Configuration : gères plusieurs classes pour le fonctionnement de l’application (Struts, Feign, etc…).
* Beans : contient les beans de l’application.

L'architecture pour la partie Front de l’application est la suivante :

* Controller : fonctionne comme RestController pour intercepter les requêtes.
* Dao : stock les méthodes pour récupérer les informations de la base de données, fonctionne avec JPA pour simplifier les requêtes.
* Model : contient les beans.
* *Les modules*

* Points particuliers
* Gestion des logs

**Log4j2 :** Trace écrite dans le fichier logInfo.txt

**Niveau des logs :** WARN

* Fichiers de configuration
* *Application web*

struts.xml /

* Ressources

**readme.md**

* Environnement de développement

**IntelliJ IDEA**

* Procédure de packaging / livraison

**Apache Maven**

* Glossaire

|  |  |
| --- | --- |
| PostgreSQL | *PostgreSQL est un système de gestion de base de données relationnelle et objet* |
| Tomcat | *Apache Tomcat est un conteneur web libre de servlets et JSP* |
| Log4j2 | *Trace des informations de plusieurs niveaux d’importances* |
| IntelliJ IDEA | IntelliJ IDEA également appelé « IntelliJ », « IDEA » ou « IDJ » est un environnement de développement intégré de technologie Java destiné au développement de logiciels informatiques |