

# Architecture technique (JM)

Dématérialisation d'un processus de paiement

COMETS Jean-Marie  
DELMARRE Adrian  
REYNOLDS Nicolas  
TURPIN Pierre

28 octobre 2014

## Table des matières

<b>1</b>	<b>Dialogue bornes</b>	<b>3</b>
1.1	Comparaison des solutions de balance de charge . . . . .	3
<b>2</b>	<b>BDD</b>	<b>3</b>
2.1	Comparaison des solutions de base de données libres . . . . .	3
<b>3</b>	<b>Gestion utilisateur / entreprise / commerçant</b>	<b>4</b>
<b>4</b>	<b>Gestion SI</b>	<b>4</b>

## 1 Dialogue bornes

Le dialogue avec les bornes peut être considérée comme l'activité la plus coûteuse, en visant un quart du marché des titres restaurants, soit 1M d'utilisateurs, on peut estimer la charge maximale à 80k connexions simultanées (80% des utilisateurs le midi au maximum, répartis par tranches de 10%).

Les bornes sont reliées à un serveur de dialogue bornes *B*, s'occupant d'effectuer les opérations spécifiques aux bornes (transaction, état, etc...). Ce serveur devant gérer une charge conséquente, on lui appliquera une **balance de charge**, distribuant entre deux autres serveurs équivalents. Chacun de ces serveurs sera installé sur une instance séparée.

Pour la solution utilisée pour le dialogue, on implémentera une solution spécifique, au vu de la spécificité du problème. La communication sera sur la base du TCP, pour garantir la fiabilité du transport.

### 1.1 Comparaison des solutions de balance de charge

On pourra utiliser *Nginx* pour la balance de charge, qui est une solution libre, performante et stable.

Une solution concurrente est *Apache*, mais à l'inconvénient d'être lourde et moins rapide au vu des dernières comparaisons.

## 2 BDD

On utilisera une solution de base de données libre car peu coûteuse, en préférant une solution relationnelle pour profiter de la conception préalable et déduire la structure interne.

Les trois serveurs de dialogue bornes auront chacun une réplication de la base de données, pour limiter davantage les accès réseau et ainsi diminuer la latence de réponse aux bornes en période de charge conséquente.

L'accès à la base de données **maître**, centralisant les répliques, sera limitée aux différentes machines comprenant des applications Aventix, via des règles firewall. L'accès aux répliques de base de données sera entièrement fermé aux connexions externes.

### 2.1 Comparaison des solutions de base de données libres

Deux solutions concurrentes se détachent :

**MySQL** actuellement maintenu par **Oracle**, c'est une solution ayant l'avantage d'être relativement simple à mettre en place.

**PostgreSQL** toujours un projet 100% libre, c'est une solution plus complexe à mettre en place, mais les dernières comparaisons la mettent en pole position au niveau de la performance au niveau de la distribution de requêtes (architecture maître-esclave, soit celle-ci).

### **3 Gestion utilisateur / entreprise / commerçant**

Chaque application de gestion sera installée sur une instance séparée, et ne devra pas supporter une charge trop conséquente. Le serveur utilisé devra gérer des connexions HTTP régulières, on choisira parmi les solutions libres encore une fois, pour réduire le cout de la solution. Le développement de l'interface demandera cependant un investissement non négligeable, vu que c'est la visibilité d'Aventix qui est en jeu.

L'accès à ces applications sera ouvert en HTTP pour l'intégralité d'internet, vu que l'accès sera global. On considérera toutefois la limitation de l'accès à l'application de gestion commerçant dans le cas d'un contrat conséquent avec un unique commerçant.

### **4 Gestion SI**

L'application Aventix sera installée sur une instance séparée, dont l'accès sera limité par un firewall plus restrictif, spécifiant de n'accepter que les connexions spécifiques au réseau d'Aventix.