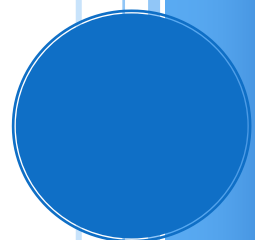


*Dossier d'architecture d'une plateforme de passage  
d'ordres boursiers*

Ce document contient la définition des différentes architectures pour la mise  
en place d'un système de passages d'ordre boursier à l'échelle nationale

Ahmed Ayadi – Yann Bondue - Nathan Kruck - Nolan Potier – SI5

07/02/2014



Contraintes exprimées.....	3
1. Besoins techniques.....	3
2. Besoins métier.....	3
3. Contraintes structurantes.....	3
i. Architecture centralisée répartie .....	3
ii. Qualité de service.....	4
Contraintes existantes.....	5
1. Contraintes organisationnelles .....	5
2. Contraintes liées à l'environnement technique .....	5
Architecture applicative .....	6
1. Description fonctionnelle .....	6
2. Description détaillée des blocs applicatifs .....	7
i. Stock information : .....	7
ii. Transaction System:.....	8
iii. Account Information : .....	9
iv. Web application:.....	10
v. Application Trader: .....	11
vi. Application Courtier:.....	12
vii. Gros Client: .....	13
viii. Agence Régionale : .....	14
ix. Agence Locale: .....	15
3. Cartographie applicative des flux .....	16
4. Matrice des flux.....	17
Architecture logicielle.....	20
1. Stock Information .....	20
i. Couches : .....	20
ii. Tiers : .....	20
2. Transaction System : .....	21
i. Couches : .....	21
ii. Tiers : .....	21
3. Account information : .....	22
i. Couches : .....	22
ii. Tiers : .....	22
4. Web Application: .....	23
i. Couches : .....	23
ii. Tiers : .....	23

5.	Application Trader :	24
i.	Couche :	24
ii.	Tiers :	24
6.	Application Courtier :	25
i.	Couches :	25
ii.	Tiers :	25
7.	Agence Régionale :	26
i.	Couches :	26
ii.	Tiers :	26
8.	Agence Locale :	27
Architecture technique		28
1.	Bloc applicatif « Stock information »	28
2.	Bloc applicatif « Transaction system »	29
3.	Bloc applicatif « Account information »	30
4.	Bloc applicatif « Web application »	31
5.	Bloc applicatif « Application trader »	32
6.	Bloc applicatif « Application courtier »	33
7.	Bloc applicatif « Agence régionale »	34
8.	Bloc applicatif « Agence locale »	35

# CONTRAINTES EXPRIMEES

## 1. Besoins techniques

### Acteurs :

- *Courtiers* : Accès au service via la Web Application.
- *Clients individuels* : Connexion avec des PC.
- *Clients mobiles* : Clients légers avec connexion très rapide : accès direct aux cours.
- *“Gros clients”* : Contrainte critique ; délai d'exécution des transactions.
- *Clients traditionnels* : pas de contraintes techniques, leurs transactions sont assurées à travers les agences régionales.
- *Les courtiers de la banque* vont assurer les transactions des clients traditionnels.
- *Agents “à la corbeille”* : accès direct à la centrale.

## 2. Besoins métier

- Fournir des outils d'aide à la décision.
- Fournir un outil de consultation de l'état du marché aussi proche du direct que possible.
- Offrir la possibilité de passer des ordres le plus rapidement possible pour les traders.
- Offrir la possibilité de passer des ordres dans toutes les agences du réseau.
- Offrir un web-service aux courtiers pour leur donner un accès à nos fonctionnalités.
- Fournir une interface riche à nos clients les plus fortunés.

## 3. Contraintes structurantes

### i. Architecture centralisée répartie

Architecture répartie dans les agences régionales et locales qui est centralisée pour l'aspect Business Intelligence dans l'agence centrale.

- Agence centrale (Paris) :
  - Reliée directement à la place de marché
  - Obtenir les actualités du marché (carnets d'ordre)
  - Doit recevoir tous les ordres des agences régionales
  - Passe les ordres sur le marché
  - Mettre à jour le carnet d'ordres
  - Stocker les historiques d'ordres
- Agences régionales :
  - Reliées à l'agence centrale
  - Transmettent les ordres à cette dernière
- Agences locales :
  - Reliées à l'agence régionale la plus proche
  - Transmettent les ordres à cette dernière

## ii. Qualité de service

La qualité de service dépendra de la localisation des utilisateurs, en effet l'agence centrale parisienne abrite l'ensemble des serveurs principaux et fournit un service hautement disponible grâce à différents mécanismes détaillés dans l'architecture physique du système. Les relais régionaux quant à eux assurent un service dont les nécessités de qualité sont inférieures puisqu'elles ne concernent qu'une partie moins importante des volumes. Il en va de même pour les agences locales à une plus grande échelle, qui nécessitent un service minimal.

## CONTRAINTES EXISTANTES

### 1. Contraintes organisationnelles

Pour les développements d'application spécifiques, nous utilisons les méthodes agiles (Scrum) en nous appuyant sur des équipes de développeurs expérimentés, certes plus onéreuses, mais plus efficaces.

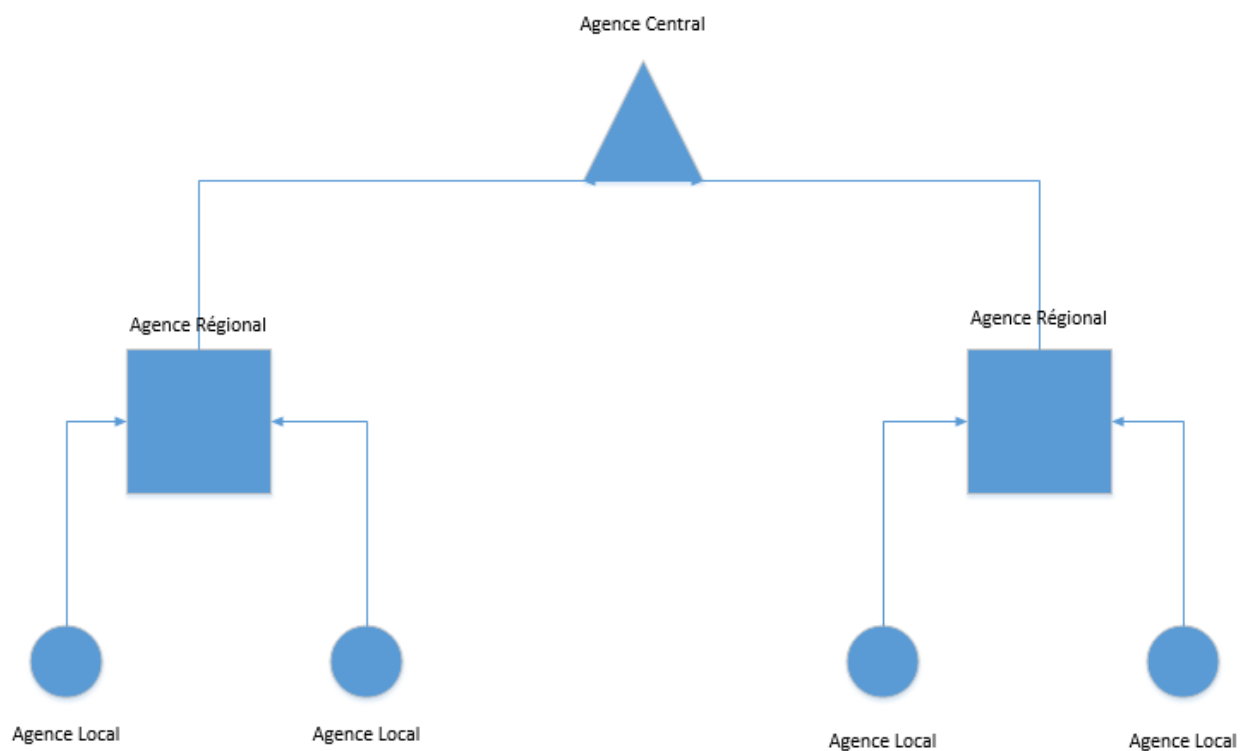
En ce qui concerne la maîtrise d'ouvrage, nous appliquons la méthode ATAM, conduite conjointement par l'architecte et le chef de projet, afin d'évaluer la qualité des composants.

### 2. Contraintes liées à l'environnement technique

Ce projet s'inscrit dans une dynamique de création d'une infrastructure nouvelle, de ce fait les contraintes liées à l'environnement technique sont inexistantes.

# ARCHITECTURE APPLICATIVE

## 1. Description fonctionnelle



3 blocs fonctionnels : Architecture répartie dans les agences régionales et locales qui est centralisée pour l'aspect Business Intelligence dans l'agence centrale.

## 2. Description détaillée des blocs applicatifs

### i. Stock information :

Le bloc Stock information est une brique cruciale de l'architecture. Il s'agit de la composante qui interroge le marché pour obtenir son état en direct et en informer le reste du système, en vue du stockage des informations et de leur utilisation dans notre contexte applicatif.

#### 1. Description fonctionnelle :

<b>Objectifs</b>	Obtenir les informations du marché et passer des ordres de la manière la plus rapide possible, afin d'être aussi proche du direct que possible
<b>Classification</b>	Critique, la gestion des informations du marché constitue la base même de notre architecture
<b>Entrées / Sorties</b>	Entrées : <ul style="list-style-type: none"> <li>- Flux d'ordres provenant du marché</li> <li>- Flux d'ordres provenant du bloc Transaction Information</li> </ul> Sorties : <ul style="list-style-type: none"> <li>- Flux de passage d'ordres provenant du bloc Transaction Information vers le marché</li> <li>- Flux de stockage de données vers une base de données du marché</li> <li>- Flux d'envoi d'informations d'ordres vers les agents à la corbeille</li> </ul>
<b>Fonctions</b>	<ul style="list-style-type: none"> <li>- Passer un Ordre</li> <li>- Récupérer les Ordres</li> </ul>
<b>Classes concepts gérées</b>	<ul style="list-style-type: none"> <li>- Lecteur et parseur d'information</li> <li>- Transmetteur d'information</li> </ul>

#### 2. Description applicative et technique

<b>Développement spécifique ou progiciel</b>	Développement spécifique
<b>Volume traité</b>	Millions de transactions/jour
<b>Disponibilité</b>	Très haute disponibilité (> 99.9%) (La fermeture de la bourse permet d'assurer une maintenance quotidienne)
<b>Fiabilité, niveau de service, performances attendues</b>	Nécessité d'être aussi proche du marché que possible, et de permettre des transactions en temps réel
<b>Matériel</b>	IBM CICS Transaction Server (Système permettant d'effectuer des transactions avec une très grande économie de moyen, très utilisé dans la finance)
<b>Système d'exploitation</b>	z/OS (Système accompagnant le serveur)
<b>SGBDR ou SGF</b>	Oracle Database (Meilleur support de grands volumes de transaction, et est porté sur notre système d'exploitation)
<b>Middleware</b>	Terminal Bloomberg
<b>Types de sites concernés</b>	Centrale (Paris)
<b>Sécurité</b>	Haute sécurité, le passage d'ordres requiert une authentification



## ii. Transaction System:

Le bloc Transaction System reçoit les données de transaction avant qu'elles ne soient passées afin d'en identifier la source et de s'assurer qu'elle est autorisée à la passer.

### 1. Description fonctionnelle :

<b>Objectifs</b>	Faire le lien entre les données de client et les données de transaction pour contrôler leur authenticité avant de les transmettre au service de passage d'ordre
<b>Classification</b>	Critique, il ne faut pas permettre à des utilisateurs non autorisés à passer des ordres
<b>Entrées / Sorties</b>	Entrées : <ul style="list-style-type: none"> <li>- Flux de transactions provenant des clients</li> <li>- Données d'autorisation associées à un client provenant du bloc Account Information</li> </ul> Sorties : <ul style="list-style-type: none"> <li>- Flux de passage d'ordres provenant de nos clients vers Stock Information</li> <li>- Demandes d'authentification vers le bloc Account Information</li> </ul>
<b>Fonctions</b>	<ul style="list-style-type: none"> <li>- Contrôler la validité des données</li> <li>- Transmettre les ordres</li> </ul>
<b>Classes concepts gérées</b>	<ul style="list-style-type: none"> <li>- Lecteur et contrôleur d'information</li> <li>- Transmetteur d'information</li> </ul>

### 2. Description applicative et technique :

<b>Développement spécifique ou progiciel</b>	Développement spécifique
<b>Volume traité</b>	Millions de transactions/jour
<b>Disponibilité</b>	Très haute disponibilité (> 99.9%) (La fermeture de la bourse permet d'assurer une maintenance quotidienne)
<b>Fiabilité, niveau de service, performances attendues</b>	Doit pouvoir gérer un grand nombre de connexions simultanées
<b>Matériel</b>	IBM CICS Transaction Server
<b>Système d'exploitation</b>	z/OS
<b>SGBDR ou SGF</b>	Microsoft SQL server
<b>Middleware</b>	N/A
<b>Types de sites concernés</b>	Centrale (Paris)
<b>Sécurité</b>	Haute sécurité, les applications qui font appel à lui doivent être clairement identifiées

### iii. Account Information :

Le bloc Account Information gère les comptes utilisateurs qui utilisent notre architecture logicielle, elle permet principalement l'enregistrement de clients.

#### 1. Description fonctionnelle:

<b>Objectifs</b>	Assurer la persistance des informations clientes et permettre de vérifier leur authenticité
<b>Classification</b>	Critique, la gestion d'utilisateurs est cruciale
<b>Entrées / Sorties</b>	Entrées : <ul style="list-style-type: none"> <li>- Flux d'informations clientes à vérifier provenant de Transaction System</li> <li>- Données clientes provenant serveur web pour la création des comptes</li> </ul> Sorties : <ul style="list-style-type: none"> <li>- Validations/Rejets d'informations d'identification</li> </ul>
<b>Fonctions</b>	- Contrôler la validité des données clientes
<b>Classes concepts gérées</b>	- CRUD

#### 2. Description applicative et technique:

<b>Développement spécifique ou progiciel</b>	Développement spécifique (partie base de données assurée par Oracle)
<b>Volume traité</b>	Milliers de transactions/jour
<b>Disponibilité</b>	Très haute disponibilité (> 99%) (La fermeture de la bourse permet d'assurer une maintenance quotidienne)
<b>Fiabilité, niveau de service, performances attendues</b>	Gérer un grand nombre d'authentification (milliers/jour), stocker un grand nombre (1 compte par client) d'informations client
<b>Matériel</b>	Oracle SPARC T4-1 (serveur dédié de base de données hautes performances)
<b>Système d'exploitation</b>	Oracle Solaris (OS de gestion de base de données couplé avec SPARC)
<b>SGBDR ou SGF</b>	SGBDR Oracle
<b>Middleware</b>	N/A
<b>Types de sites concernés</b>	Centrale (Paris)
<b>Sécurité</b>	Très haute sécurité, la base de données contient des informations sensibles et confidentielles

## iv. Web application:

Le bloc Web application représente le site de notre agence et permet aux clients non-professionnels de créer un compte et de passer leurs ordres personnels depuis la plate-forme.

## 1. Description fonctionnelle:

<b>Objectifs</b>	Proposer les web services pour les différentes applications clientes légères
<b>Classification</b>	Important, touche le plus grand nombre de clients à petite échelle
<b>Entrées / Sorties</b>	Entrées : <ul style="list-style-type: none"> <li>- Flux d'informations clientes de créations de compte et de passage d'ordres</li> <li>- Reçoit les données du marché via Stock Information</li> </ul> Sorties : <ul style="list-style-type: none"> <li>- Transmission des informations à vérifier vers Transaction System</li> <li>- Transmission des informations clientes vers Account Information</li> <li>- Transmission de confirmations et d'informations aux utilisateurs</li> </ul>
<b>Fonctions</b>	<ul style="list-style-type: none"> <li>- Créer un compte utilisateur</li> <li>- Passer un ordre</li> <li>- Récupérer les informations en base de données</li> <li>- Transmettre des informations aux clients</li> </ul>
<b>Classes concepts gérées</b>	<ul style="list-style-type: none"> <li>- Interfaces Web Services</li> </ul>

## 2. Description applicative et technique :

<b>Développement spécifique ou progiciel</b>	Développement spécifique
<b>Volume traité</b>	Milliers de transactions/jour
<b>Disponibilité</b>	Très haute disponibilité (> 99.99%) (La fermeture de la bourse permet d'assurer un service dégradé de nuit)
<b>Fiabilité, niveau de service, performances attendues</b>	Assurer le passage d'ordres pour des clients individuels à petite échelle, avec un niveau d'exigence plus faible que les clients privilégiés
<b>Matériel</b>	Dell PowerEdge M520 (Serveur haute-performance pour applications web)
<b>Système d'exploitation</b>	Microsoft Windows Server 2012 R2 (Permet de déployer des web services)
<b>SGBDR ou SGF</b>	N/A
<b>Middleware</b>	N/A
<b>Types de sites concernés</b>	Centrale (Paris)
<b>Sécurité</b>	Les applications utilisant les web services doivent être clairement identifiées et autorisées

## v. Application Trader:

Le bloc applicatif Trader permet aux agents à la corbeille d'agir directement au plus près du marché, en limitant tous les intermédiaires et en leur fournissant une connexion très haut débit.

## 1. Description fonctionnelle:

<b>Objectifs</b>	Assurer un passage d'ordre en temps réel et être informé en temps réel des fluctuations du marché
<b>Classification</b>	Important, il s'agit de la composante qui permet à la banque d'assurer la plus grande partie de ses transactions à grande échelle
<b>Entrées / Sorties</b>	Entrées : <ul style="list-style-type: none"> <li>- Reçoit les données du marché via Stock Information</li> <li>- Reçoit les données de transactions de l'agent à la corbeille</li> </ul> Sorties : <ul style="list-style-type: none"> <li>- Transmission des ordres à Stock Information</li> </ul>
<b>Fonctions</b>	<ul style="list-style-type: none"> <li>- Passer un ordre</li> <li>- Recevoir les informations du marché</li> </ul>
<b>Classes concepts gérées</b>	<ul style="list-style-type: none"> <li>- Récepteur d'informations</li> <li>- Émetteur d'informations</li> <li>- CRUD</li> </ul>

## 2. Description applicative et technique :

<b>Développement spécifique ou progiciel</b>	Développement spécifique
<b>Volume traité</b>	Millions de transactions/jour
<b>Disponibilité</b>	Très haute disponibilité (> 99.999%) (La fermeture de la bourse permet d'assurer une maintenance de nuit)
<b>Fiabilité, niveau de service, performances attendues</b>	Nécessité d'être aussi proche du marché que possible, et de permettre des transactions en temps réel
<b>Matériel</b>	Dell PowerEdge M420 (Idéal pour assurer un grand nombre de petites transactions rapides)
<b>Système d'exploitation</b>	Microsoft Windows Server 2012 R2 (Permet un déploiement rapide)
<b>SGBDR ou SGF</b>	N/A
<b>Middleware</b>	N/A
<b>Types de sites concernés</b>	Centrale (Paris)
<b>Sécurité</b>	Seuls des terminaux réservés peuvent accéder au serveur

## vi. Application Courtier:

Le bloc applicatif Courtier permet de proposer aux courtiers un service de passage d'ordre afin qu'ils puissent fournir un service équivalent à leur clientèle.

## 1. Description fonctionnelle:

<b>Objectifs</b>	Assurer un passage d'ordre pour les courtiers externes à notre réseau
<b>Classification</b>	Important, ce bloc permet à la banque de gérer tous ses clients externes
<b>Entrées / Sorties</b>	Entrées : <ul style="list-style-type: none"> <li>- Reçoit les données du marché via Stock Information</li> <li>- Reçoit les données de transactions du courtier</li> </ul> Sorties : <ul style="list-style-type: none"> <li>- Transmission des ordres à Stock Information</li> </ul>
<b>Fonctions</b>	Passer un ordre <ul style="list-style-type: none"> <li>- Recevoir les informations du marché</li> </ul>
<b>Classes concepts gérées</b>	<ul style="list-style-type: none"> <li>- Récepteur d'informations</li> <li>- Émetteur d'informations</li> </ul>

## 2. Description applicative et technique :

<b>Développement spécifique ou progiciel</b>	Développement spécifique
<b>Volume traité</b>	Milliers de transactions/jour
<b>Disponibilité</b>	Très haute disponibilité (> 99%) (La fermeture de la bourse permet d'assurer une maintenance de nuit)
<b>Fiabilité, niveau de service, performances attendues</b>	Garantir le passage des ordres en un temps minimal
<b>Matériel</b>	Dell PowerEdge M520 (Serveur haute-performance pour applications web)
<b>Système d'exploitation</b>	Microsoft Windows Server 2012 R2 (Permet de déployer des web services)
<b>SGBDR ou SGF</b>	N/A
<b>Middleware</b>	N/A
<b>Types de sites concernés</b>	Centrale (Paris)
<b>Sécurité</b>	Seuls des courtiers identifiés peuvent accéder au serveur

## vii. Gros Client:

Le bloc applicatif Gros Client est un software déployé auprès des agences de clients qui leur donne un accès privilégié pour passer des ordres à grande échelle.

## 1. Description fonctionnelle :

<b>Objectifs</b>	Assurer un passage d'ordre de grands volumes et montants en un temps minimal
<b>Classification</b>	Important, ce bloc permet à la banque de gérer tous ses clients principaux
<b>Entrées / Sorties</b>	Entrées : <ul style="list-style-type: none"> <li>- Reçoit les données du marché via Stock Information</li> <li>- Reçoit les données de transaction du client</li> <li>- Reçoit les vérifications d'authentification de Account Information</li> </ul> Sorties : <ul style="list-style-type: none"> <li>- Transmission des ordres à Stock Information</li> <li>- Émet des données d'authentification vers Account Information</li> </ul>
<b>Fonctions</b>	<ul style="list-style-type: none"> <li>- Passer un ordre</li> <li>- Recevoir les informations du marché</li> </ul>
<b>Classes concepts gérées</b>	<ul style="list-style-type: none"> <li>- Récepteur d'informations</li> <li>- Émetteur d'informations</li> </ul>

## 2. Description applicative et technique :

<b>Développement spécifique ou progiciel</b>	Développement spécifique
<b>Volume traité</b>	Milliers de transactions/jour
<b>Disponibilité</b>	Très haute disponibilité (> 99.9%) (La fermeture de la bourse permet d'assurer une maintenance de nuit)
<b>Fiabilité, niveau de service, performances attendues</b>	Garantir le passage des ordres en un temps minimal
<b>Matériel</b>	Dell PowerEdge M520 (Serveur haute-performance pour applications web)
<b>Système d'exploitation</b>	Microsoft Windows Server 2012 R2 (Permet de déployer des web services)
<b>SGBDR ou SGF</b>	N/A
<b>Middleware</b>	N/A
<b>Types de sites concernés</b>	Centrale (Paris)
<b>Sécurité</b>	Seuls des clients identifiés peuvent accéder au serveur

## viii. Agence Régionale :

Le bloc applicatif des agences régionales consiste en une application légère reliée à un serveur local, ce serveur communique ensuite avec les web services du bloc applicatif Web.

## 1. Description fonctionnelle :

<b>Objectifs</b>	Assurer un passage d'ordre pour les agences régionales au travers d'une application dédiée
<b>Classification</b>	Important, ce bloc permet aux agences de la banque de fournir le service à ses clients
<b>Entrées / Sorties</b>	Entrées : <ul style="list-style-type: none"> <li>- Reçoit les données du marché via Web</li> <li>- Reçoit les données de transaction de l'agence</li> </ul> Sorties : <ul style="list-style-type: none"> <li>- Transmission des ordres à Web</li> </ul>
<b>Fonctions</b>	<ul style="list-style-type: none"> <li>- Passer un ordre</li> <li>- Recevoir les informations du marché</li> </ul>
<b>Classes concepts gérées</b>	<ul style="list-style-type: none"> <li>- Récepteur d'informations</li> <li>- Émetteur d'informations</li> </ul>

## 2. Description applicative et technique :

<b>Développement spécifique ou progiciel</b>	Développement spécifique
<b>Volume traité</b>	Milliers de transactions/jour
<b>Disponibilité</b>	Haute disponibilité (> 90%) (La fermeture de la bourse permet d'assurer une maintenance de nuit)
<b>Fiabilité, niveau de service, performances attendues</b>	Garantir le passage des ordres sans véritables contraintes temporelles
<b>Matériel</b>	Dell PowerEdge T110 II (Serveur idéal pour un petit nombre de transactions et de connexions)
<b>Système d'exploitation</b>	Microsoft Windows Server 2012 R2 (Permet un déploiement simple et rapide)
<b>SGBDR ou SGF</b>	N/A
<b>Middleware</b>	N/A
<b>Types de sites concernés</b>	Régional (Province)
<b>Sécurité</b>	Seul le personnel de l'agence à accès au serveur.

## ix. Agence Locale:

Le bloc applicatif des agences locales consiste en un terminal dédié aux fonctionnalités limitées et en communication avec le serveur de l'agence régionale afférente.

## 1. Description fonctionnelle :

<b>Objectifs</b>	Assurer un passage d'ordre pour les agences locales au travers d'une application dédiée
<b>Classification</b>	Utile, ce bloc permet aux agences de la banque de fournir le service à ses clients les moins avertis
<b>Entrées / Sorties</b>	Entrées : <ul style="list-style-type: none"> <li>- Reçoit les données du marché via l'agence régionale afférente</li> <li>- Reçoit les données de transaction de l'opérateur</li> </ul> Sorties : <ul style="list-style-type: none"> <li>- Transmission des ordres à l'agence régionale</li> </ul>
<b>Fonctions</b>	Passer un ordre <ul style="list-style-type: none"> <li>- Recevoir les informations du marché</li> </ul>
<b>Classes concepts gérées</b>	<ul style="list-style-type: none"> <li>- Récepteur d'informations</li> <li>- Émetteur d'informations</li> </ul>

## 2. Description applicative et technique :

<b>Développement spécifique ou progiciel</b>	Développement spécifique
<b>Volume traité</b>	Milliers de transactions/jour
<b>Disponibilité</b>	Haute disponibilité (> 90%) (La fermeture de la bourse permet d'assurer une maintenance de nuit)
<b>Fiabilité, niveau de service, performances attendues</b>	Garantir le passage des ordres sans aucunes contraintes temporelles
<b>Matériel</b>	Dell OptiFlex 7010 (Ordinateur fourni au personnel des agences)
<b>Système d'exploitation</b>	Microsoft Windows 8.1 (Suffisant pour l'usage réalisé)
<b>SGBDR ou SGF</b>	N/A
<b>Middleware</b>	N/A
<b>Types de sites concernés</b>	Local
<b>Sécurité</b>	Seul le personnel à accès à l'ordinateur



### 3. Cartographie applicative des flux

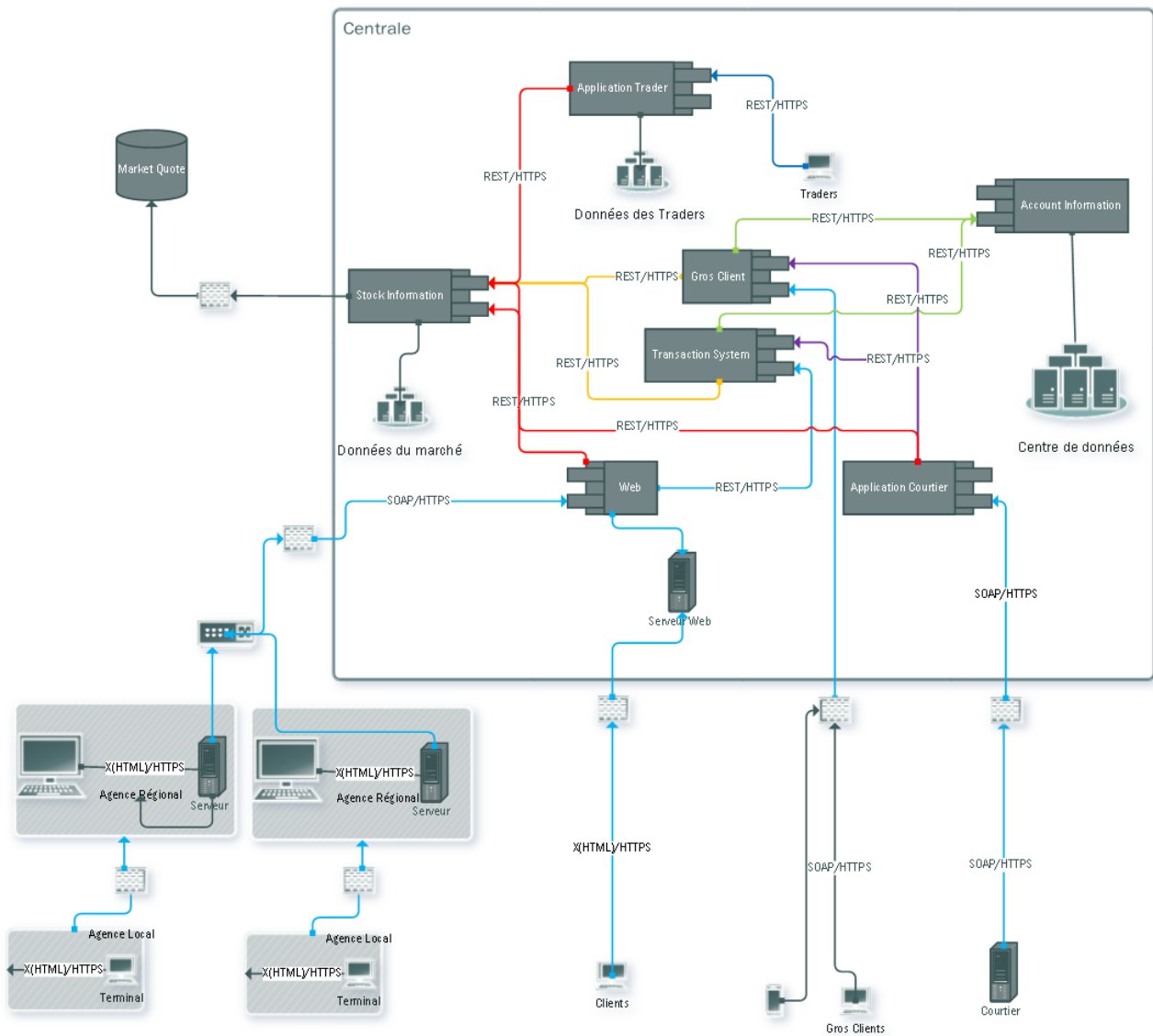


Figure 1 : Cartographie applicative des flux

## 4. Matrice des flux

#	Depuis	Vers	Support	TP/Batch	Format message	Protocole de Transport	Description
1	Navigateur client de l'agence régional	Serveur agence régionale	Ethernet	TP	(x)HTML	HTTPS	Connexion au portail de l'agence régional
2	Navigateur client de l'agence local	Serveur agence régionale	Internet	TP	(x)HTML	HTTPS	Connexion au portail de l'agence régional
3	Serveur agence régional	Serveur web services agence centrale	Internet	Batch	SOAP	HTTPS	Utilisation des web services de l'agence centrale
4	Serveur web services agence centrale	Transaction System	Ethernet	Batch	REST	HTTPS	Gere le passage des ordres
5	Transaction System	Stock Information	Ethernet	Batch	REST	HTTPS	Traitement des données du marché
6	Transaction System	Account Information	Ethernet	Batch	REST	HTTPS	Traitement des informations client
7	Account Information	Data Base Acces	Ethernet	Batch	REST	HTTPS	Accès à la base de données
8	Stock Information	Market Quote	Internet	Batch	?	HTTPS	Passe les ordres sur le marché

## 1. Cinématique représentative de l'utilisation du système :

UC : Afficher les flux des prix

User : Client Ordinaire

Précondition : Client Authentifié

Etape	Description
1	Le client à travers son navigateur demande l'affichage du flux des prix d'une valeur boursière choisie
2	La requête est transmise vers l'application web des clients ordinaires
3	L'application web fait un appel en SOAP/HTTPS au block stock information system
4	Le block <b>Stock Information System</b> vérifie si le prix de la valeur boursière est le plus récent (peut inclure une requête REST/HTTPS pour mettre à jours le carnet d'ordre) et répond en REST/HTTPS au block <b>Web</b> .
5	Le block <b>Web</b> répond au client en html.
6	3 jusqu'à 5 se répètent jusqu'à le client affiche autre chose.

## 2. Vision Dynamique

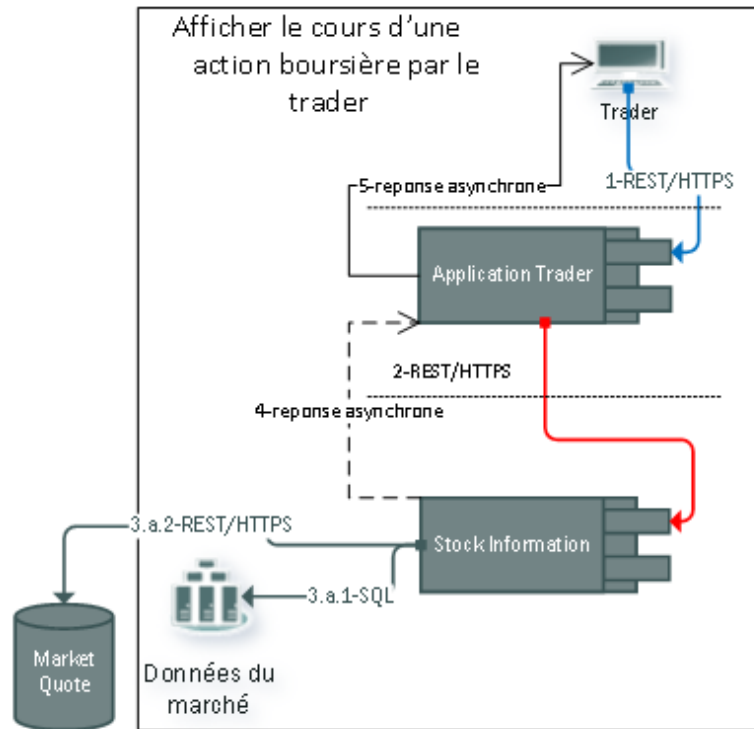


Figure 2 : Vision dynamique

# ARCHITECTURE LOGICIELLE

Après la modélisation des besoins des utilisateurs du système, on a pu déceler les exigences. Cela nous a permis de :

- Mettre le focus sur le comportement extérieur du système, indépendamment de sa conception interne.
- Décrire les besoins des utilisateurs et des parties prenantes « avec beaucoup moins d'ambiguïté ».
- Comblent les lacunes et les incohérences dans les exigences.
- Planifiez l'ordre dans lequel les caractéristiques seront élaborées.
- Utilisez les modèles de base pour les tests du système, ce qui rend une relation claire entre les critères et les exigences. Lorsque les conditions changent, cette relation permet de mettre à jour les essais correctement. Cela permet de s'assurer que le système répond aux nouvelles exigences.

## 1. Stock Information

Le logiciel implémenté va servir de parseur des données du marché. Il devra pouvoir rendre persistantes ces données et fournir des services de lecture et écriture sur ces données. Il doit aussi passer des ordres sur le marché.

Le programme étant exécuté sur le système d'exploitation z/OS et nécessitant de très bonne performance en terme de rapidité sera implémenté avec le langage C++.

### i. Couches :

Présentation	<b>N/A : aucune interaction avec l'utilisateur</b>
Contrôleur	Doit contrôler l'appel de services, et doit contrôler que le carnet d'ordre présent dans la base de données est à jours.
Services	Web services REST <ul style="list-style-type: none"> <li>- Récupérer un carnet d'ordre</li> <li>- Passer un ordre</li> </ul> <b>Performance</b> : Très performant <b>Security</b> : Contraintes sécurité fortes <b>Robustesse</b> : Très robuste même avec un comportement dégradé du system
Domaine	<b>Objets métiers :</b> <ul style="list-style-type: none"> <li>- Carnet d'ordres (liste d'ordres) ; Utilisation de la librairie STL</li> <li>- Ordre Trader Ordre</li> <li>- Gros Client</li> <li>- Ordre</li> </ul> <b>Fourniture des moyens d'accès à l'information :</b> Oracle Call Interface (OCI) est une API fournie par Oracle permettant aux développeurs de créer des applications en utilisant des appels C++ bas niveau afin d'accéder à des bases de données Oracle.
Persistance	Librairie OCILIB encapsule OCI afin de fournir une interface beaucoup plus simple à coder et facilite la maintenance.

### ii. Tiers :

Modèle 2-tiers.

1 serveur Logique : IBM CICS z/OS : application C++

1 serveur Données : Oracle

## 2. Transaction System :

Le logiciel implémenté va servir de contrôleur afin d'autoriser les ordres avant que ceux-ci ne soit passé au Stock Information.

### i. Couches :

Présentation	<b>N/A : aucune interaction avec l'utilisateur</b>
Contrôleur	Doit contrôler l'appel de services, Contrôler la validité des données et appelant les services de bloc Account Information
Services	Web services REST - Passer un ordre (ordre, client) <b>Performance</b> : Très performant <b>Security</b> : Contraintes sécurité fortes <b>Robustesse</b> : Très robuste même avec un comportement dégradé du system ; doit assurer des contraintes de transactions sur les ordres qu'il traite.
Domaine	<b>Objets métiers :</b> - Carnet d'ordres (liste d'ordres) Utilisation de la librairie STL - Ordre - Client <b>Fourniture des moyens d'accès à l'information :</b> Les accès à l'information son géré par le bloc Account Information
Persistance	La persistance des données est gérée par le bloc Account Information

### ii. Tiers :

Modèle 1-tiers.

1 serveur Logique : IBM CICS z/OS : application C++

### 3. Account information :

Le logiciel implémenté va servir à persister toutes les données relatives aux clients.

Le programme étant exécuté sur le système d'exploitation z/OS et nécessitant de très bonne performance en terme de rapidité sera implémenté avec le langage C++.

#### i. Couches :

Présentation	<b>N/A : aucune interaction avec l'utilisateur</b>
Contrôleur	Doit contrôler l'appel de services, et doit permettre de créer, rechercher, modifier et supprimer des composants d'objets métiers dans le respect des propriétés transactionnelles.
Services	Web services REST <ul style="list-style-type: none"> <li>- Créer un client</li> <li>- Supprimer un client</li> <li>- Modifier un client</li> <li>- Ajouter un ordre</li> <li>- Supprimer un ordre</li> <li>- Modifier un ordre</li> </ul> <b>Performance</b> : Très performant <b>Security</b> : Contraintes sécurité fortes <b>Robustesse</b> : Très robuste même avec un comportement dégradé du system
Domaine	<b>Objets métiers :</b> <ul style="list-style-type: none"> <li>- Carnet d'ordres (liste d'ordres) ; Utilisation de la librairie STL</li> <li>- Ordre</li> <li>- Client</li> <li>- Gros Client</li> </ul> <b>Fourniture des moyens d'accès à l'information :</b> Oracle Call Interface (OCI) est une API fournie par Oracle permettant aux développeurs de créer des applications en utilisant des appels C++ bas niveau afin d'accéder à des bases de données Oracle.
Persistance	Librairie OCILIB encapsule OCI afin de fournir une interface beaucoup plus simple à coder et facilite la maintenance.

#### ii. Tiers :

Modèle 1-tiers.

1 serveur Données : Oracle SPARC T4-1

#### 4. Web Application:

Le logiciel va implémenter tous les web services et l'application web destiné aux clients. Les web services seront utilisés par l'application web dédié aux clients individuel ainsi que par les applications web des agences régionale.

Le programme étant essentiellement axé web services et application web il sera implémenté en C#/ASP .NET.

##### i. Couches :

<b>Présentation</b>	<b>L'application fournit un client léger implémenté en ASP .net. Cette application dédiée aux clients individuels fournit les services de passage d'ordre.</b>
<b>Contrôleur</b>	Doit contrôler l'appel des services, doit contrôler l'authentification des clients, la cinématique des écrans et les sessions utilisateur.
<b>Services</b>	Web services SOAP <ul style="list-style-type: none"> <li>- S'authentifier</li> <li>- Récupérer carnet d'ordre</li> <li>- Passer un ordre</li> <li>- Annuler un ordre</li> <li>- Modifier un ordre</li> <li>- Créer un compte utilisateur</li> <li>- Modifier un compte utilisateur</li> </ul>
<b>Domaine</b>	<b>Objets métiers :</b> <ul style="list-style-type: none"> <li>- Carnet d'ordres (liste d'ordres)</li> </ul> Utilisation de la librairie STL <ul style="list-style-type: none"> <li>- Ordre</li> <li>- Client</li> </ul> <b>Fourniture des moyens d'accès à l'information :</b> Les accès à l'information son géré par le bloc Account Information
<b>Persistance</b>	La persistance des données est gérée par le bloc Account Information

##### ii. Tiers :

Modèle 3-tiers.

1 serveur Client : Client (ASP .NET)

1 serveur Logique : Web Services (C# .NET)

1 Serveur Données : Données (Account Information)



## 5. Application Trader :

Le logiciel permet aux agents à la corbeille de passer leurs ordres directement au plus près du marché, en limitant toute autre traitement. Il va servir de contrôleur d'autorisation de passage d'ordres avant que ceux-ci ne soit passé au Stock Information.

### i. Couche :

Présentation	<b>Doit fournir une ergonomie acceptable par les agents à la Corbeil.</b> <b>L'application fournit un client léger implémenté en ASP .net. Cette application dédiée aux traders fournit les services de passage d'ordre et de réception du flux du marché.</b>
Contrôleur	Doit contrôler l'appel des services, et doit permettre de créer, rechercher, modifier et supprimer des composants d'objets métiers dans le respect des propriétés transactionnelles.
Services	Web services REST <ul style="list-style-type: none"> <li>- Récupérer un carnet d'ordre</li> <li>- Passer un ordre (ordre, Trader)</li> </ul> <b>Performance</b> : Très performant <b>Security</b> : Contraintes sécurité fortes <b>Robustesse</b> : Très robuste même avec un comportement dégradé du system ; doit assurer des contraintes de transactions sur les ordres qu'il traite.
Domaine	<b>Objets métiers :</b> <ul style="list-style-type: none"> <li>- Carnet d'ordres (liste d'ordres) ; Utilisation de la librairie STL</li> <li>- Ordre Trader</li> <li>- Trader</li> </ul> <b>Fourniture des moyens d'accès à l'information :</b> Oracle Call Interface (OCI) est une API fournie par Oracle permettant aux développeurs de créer des applications en utilisant des appels C++ bas niveau afin d'accéder à des bases de données Oracle.
Persistance	Librairie OCILIB encapsule OCI afin de fournir une interface beaucoup plus simple à coder et facilite la maintenance.

### ii. Tiers :

Modèle 2-tiers.

1 serveur Logique : Windows Server 2012 R2

1 serveur Données : Microsoft SQL Server 2012

## 6. Application Courtier :

Le logiciel implémenté est dédié pour les courtiers externes de l'entreprise. Il doit fournir des services qui pourront être utilisé par des applications externes permettant de passer des ordres et récupérer des données du marché.

### i. Couches :

<b>Présentation</b>	<b>L'application fournit un client léger implémenté en ASP .net. Cette application dédiée aux courtiers fournit les services de passage d'ordre et de réception du flux du marché.</b>
<b>Contrôleur</b>	Doit contrôler l'appel de services
<b>Services</b>	Web services SOAP C# <ul style="list-style-type: none"> <li>- Récupérer un carnet d'ordre</li> <li>- Passer un ordre</li> </ul>
<b>Domaine</b>	<b>Objets métiers :</b> <ul style="list-style-type: none"> <li>- Carnet d'ordres (liste d'ordres)</li> </ul> Utilisation de la librairie STL <ul style="list-style-type: none"> <li>- Ordre</li> </ul> <b>Fourniture des moyens d'accès à l'information :</b> Les accès à l'information son géré par le bloc Stock Information
<b>Persistance</b>	La persistance des données est gérée par le bloc Stock Information

### ii. Tiers :

Modèle 1-tiers.

1 serveur Application : Client + Web Services (C#)

## 7. Agence Régionale :

Le logiciel implémenté est dédié pour l'agence régionale et pour les agences locales reliées à celle-ci. Il doit fournir des services qui pourront être utilisés par les agents de l'entreprise.

### i. Couches :

<b>Présentation</b>	<b>L'application fournit un client léger implémenté en ASP .net. Cette application dédiée aux agents fournit les services de passage d'ordre et de réception du flux du marché.</b>
<b>Contrôleur</b>	Doit contrôler l'appel de services
<b>Services</b>	Web services SOAP C# <ul style="list-style-type: none"> <li>- Récupérer un carnet d'ordre</li> <li>- Passer un ordre</li> </ul>
<b>Domaine</b>	<b>Objets métiers :</b> <ul style="list-style-type: none"> <li>- Carnet d'ordres (liste d'ordres)</li> </ul> Utilisation de la librairie STL <ul style="list-style-type: none"> <li>- Ordre</li> </ul> <b>Fourniture des moyens d'accès à l'information :</b> Les accès à l'information sont gérés par le bloc Stock Information
<b>Persistance</b>	La persistance des données est gérée par le bloc Stock Information

### ii. Tiers :

Modèle 3-tiers.

1 serveur Client : Client léger (ASP .NET)

1 Serveur Logique : Web Services (Web Application)

1 Serveur Données : Données (Account Information)

## 8. Agence Locale :

Le bloc applicatif Agence Locale n'a pas de couche logicielle dédiée. Elle utilise la couche logicielle du bloc applicatif Agence Régionale. Le personnel de l'entreprise utilisera le client léger de l'Agence Régionale à partir d'un poste de travail.

# ARCHITECTURE TECHNIQUE

## 1. Bloc applicatif « Stock information »

Architecture physique	<p><b>Matériel utilisé : IBM CICS Transaction Server (Système permettant d'effectuer des transactions avec une très grande économie de moyen, très utilisé dans la finance) avec un système d'exploitation z/OS fourni avec le serveur. On ajoute un SGBD Oracle Database qui est porté sur le système d'exploitation et qui permet la bonne gestion de grands volumes de transactions</b></p> <p><b>Le matériel est connecté au marché par haut débit (fibre optique) et aux autres infrastructures LAN par fibre optique afin d'avoir un débit optimal. Un firewall est mis en place sur la connexion en fibre vers l'extérieur du réseau de l'entreprise pour en assurer la sécurité, ce firewall est doublé en cas de panne du premier.</b></p>
Qualité de service	<p>Très haute disponibilité (&gt; 99.9%) seulement pendant les heures d'ouverture de la bourse, que l'on étend aux périodes de pré-ouverture/pré-clôture soit de 7h30 à 17h35. Les samedis et dimanches ainsi que les nuits permettent donc d'exécuter les batchs et les mises à jour.</p> <p>De plus, pour avoir les informations du marché le plus rapidement possible il est capital que les installations soient proches de la bourse.</p> <p>La disponibilité est assurée par un couple de serveurs capables de supporter un très grand nombre de transactions par minute, le fail-over est assuré par un troisième serveur de back-up déjà configuré et le load-balancing par un KEMP LM-5300.</p>
Dimensionnement	Deux serveurs sont à prévoir, en effet si une panne survient il est impératif qu'un second serveur soit déjà configuré et prêt à être utilisé.
Architecture d'exploitation	<p>L'utilisation de l'outil IBM Tivoli Monitoring parfaitement adapté à notre matériel permet une supervision proactive de l'ensemble des données et des composants de notre serveur IBM</p> <p>Des tests de reprises sont conduits tous les mois afin de s'assurer de la bonne connaissance et de la maîtrise du processus de reprise. Une sauvegarde des données est effectuée tous les soirs.</p>
Coût	<p>Tivoli : 1000€ HT</p> <p>Oracle Database : cf constructeur</p> <p>z/OS : cf constructeur</p> <p>CICS Transaction Server : cf constructeur</p> <p>Cisco Catalyst 6500 : 1000€ (firewall)</p>

## 2. Bloc applicatif « Transaction system »

Architecture physique	<p>Matériel utilisé : IBM CICS Transaction Server (Système permettant d'effectuer des transactions avec une très grande économie de moyen, très utilisé dans la finance) avec un système d'exploitation z/OS fourni avec le serveur. On ajoute un SGBD Windows SQL Server</p> <p>Le matériel aux autres infrastructures LAN par fibre optique afin d'avoir un débit optimal.</p> <p>Un firewall est mis en place sur la connexion en fibre vers l'extérieur du réseau de l'entreprise pour en assurer la sécurité, ce firewall est doublé en cas de panne du premier.</p>
Qualité de service	<p>Très haute disponibilité (&gt; 99.9%) seulement pendant les heures d'ouverture de la bourse, que l'on étend aux périodes de pré-ouverture/pré-clôture soit de 7h30 à 17h35. Les samedis et dimanches ainsi que les nuits permettent donc d'exécuter les batch et les mises à jour.</p> <p>Ce serveur permet de transmettre les ordres au « Stock information » il est donc capital qu'il soit performant.</p> <p>La disponibilité est assurée par un couple de serveurs capables de supporter un très grand nombre de transactions par minute, le fail-over est assuré par un troisième serveur de back-up déjà configuré et le load-balancing par un KEMP LM-5300.</p>
Dimensionnement	Deux serveurs sont à prévoir, en effet si une panne survient il est impératif qu'un second serveur soit déjà configuré et prêt à être utilisé.
Architecture d'exploitation	<p>L'utilisation de l'outil IBM Tivoli Monitoring parfaitement adapté à notre matériel permet une supervision proactive de l'ensemble des données et des composants de notre serveur IBM</p> <p>Des tests de reprises sont conduits tous les mois afin de s'assurer de la bonne connaissance et de la maîtrise du processus de reprise. Une sauvegarde des données est effectuée tous les soirs.</p>
Coût	<p>Tivoli : 1000€ HT</p> <p>Oracle Database : cf constructeur</p> <p>z/OS : cf constructeur</p> <p>CICS Transaction Server : cf constructeur</p> <p>Cisco Catalyst 6500 : 1000€ (firewall)</p>

### 3. Bloc applicatif « Account information »

Architecture physique	<b>Matériel utilisé : Oracle SPARC T4-1 (serveur dédié pour les bases de données hautes performances) avec le système d'exploitation Oracle Solaris (OS de gestion de base de données couplé avec SPARC). Le serveur est couplé avec un système de SGBDR Oracle.</b>
Qualité de service	Très haute disponibilité (> 99%) seulement pendant les heures d'ouverture de la bourse, que l'on étend aux périodes de pré-ouverture/pré-clôture soit de 7h30 à 17h35. Les samedis et dimanches ainsi que les nuits permettent donc d'exécuter les batchs et les mises à jour. Ce service permet l'authentification des clients il faut donc que les données soient sécurisées (grâce à la sécurité embarquée dans le serveur).
Dimensionnement	Deux serveurs sont à prévoir, en effet si une panne survient il est impératif qu'un second serveur soit déjà configuré et prêt à être utilisé. On utilise également la technologie RAID 5 afin d'optimiser l'accès aux données ainsi que la récupération en cas de panne d'un des disques durs.
Architecture d'exploitation	Des tests de reprises sont conduits tous les deux mois afin de s'assurer de la bonne connaissance et de la maîtrise du processus de reprise. Une sauvegarde des données est effectuée tous les soirs.
Coût	Oracle SPARC T4-1 : 45000€ Oracle DB High availability : 10000€ Oracle Solaris : inclus

#### 4. Bloc applicatif « Web application »

Architecture physique	<p><b>Matériel utilisé : Dell PowerEdge M520 (Serveur haute-performance pour applications web) équipé du système d'exploitation Microsoft Windows Server 2012 R2 (Permet de déployer des web services).</b></p> <p><b>Un firewall est mis en place sur la connexion en fibre vers l'extérieur du réseau de l'entreprise pour en assurer la sécurité, ce firewall est doublé en cas de panne du premier.</b></p>
Qualité de service	Très haute disponibilité (> 99.99%) seulement pendant les heures d'ouverture de la bourse, que l'on étend aux périodes de pré-ouverture/pré-clôture soit de 7h30 à 17h35. Les samedis et dimanches ainsi que les nuits permettent donc d'exécuter les batchs et les mises à jour.
Dimensionnement	En raison du nombre de client ces serveurs doivent être en cluster de X serveurs
Architecture d'exploitation	Des tests de reprises sont conduits tous les deux mois afin de s'assurer de la bonne connaissance et de la maîtrise du processus de reprise. Une sauvegarde des données est effectuée toute les semaines.
Coût	<p>Dell PowerEdge M520 : 4500€ (HT/unitaire)</p> <p>Microsoft Windows Server 2012 R2 : inclus</p> <p>Cisco Catalyst 6500 : 1000€ (firewall)</p>



## 5. Bloc applicatif « Application trader »

Architecture physique	<b>Matériel utilisé : Serveur Dell PowerEdge M420 (Idéal pour assurer un grand nombre de petites transactions rapides) avec un système d'exploitation Microsoft Windows Server 2012 R2 (Permet un déploiement rapide).</b>
Qualité de service	Très haute disponibilité (> 99.99%) seulement pendant les heures d'ouverture de la bourse, que l'on étend aux périodes de pré-ouverture/pré-clôture soit de 7h30 à 17h35. Les samedis et dimanches ainsi que les nuits permettent donc d'exécuter les batchs et les mises à jour.
Dimensionnement	Deux serveurs sont à prévoir, en effet si une panne survient il est impératif qu'un second serveur soit déjà configuré et prêt à être utilisé.
Architecture d'exploitation	Des tests de reprises sont conduits tous les mois afin de s'assurer de la bonne connaissance et de la maîtrise du processus de reprise. Une sauvegarde des données est effectuée tous les soirs.
Coût	Dell PowerEdge M420 : 5000€ (HT/unitaire) Microsoft Windows Server 2012 R2 : inclus

## 6. Bloc applicatif « Application courtier »

<b>Architecture physique</b>	Matériel utilisé : Dell PowerEdge M520 (Serveur haute-performance pour applications web) équipé du système d'exploitation Microsoft Windows Server 2012 R2 (Permet de déployer des web services). Les courtiers sont reliés indirectement à l'agence centrale via connexion à un web service. Un firewall est également prévu pour contrôler les connexions. Un firewall est mis en place sur la connexion en fibre vers l'extérieur du réseau de l'entreprise pour en assurer la sécurité, ce firewall est doublé en cas de panne du premier.
<b>Qualité de service</b>	Très haute disponibilité (> 99%) seulement pendant les heures d'ouverture de la bourse, que l'on étend aux périodes de pré-ouverture/pré-clôture soit de 7h30 à 17h35. Les samedis et dimanches ainsi que les nuits permettent donc d'exécuter les batchs et les mises à jour.
<b>Dimensionnement</b>	Deux serveurs sont à prévoir, en effet si une panne survient il est impératif qu'un second serveur soit déjà configuré et prêt à être utilisé.
<b>Architecture d'exploitation</b>	Des tests de reprises sont conduits tous les trois mois afin de s'assurer de la bonne connaissance et de la maîtrise du processus de reprise. Une sauvegarde des données est effectuée tous les mois.
<b>Coût</b>	Dell PowerEdge M520 : 4500€ (HT/unitaire) Microsoft Windows Server 2012 R2 : inclus Firewall : Cisco Catalyst 6500 : 1000€

## 7. Bloc applicatif « Agence régionale »

<b>Architecture physique</b>	<p>Matériel utilisé : Dell PowerEdge T110 II (Serveur idéal pour un petit nombre de transactions et de connexions) équipé du système d'exploitation Microsoft Windows Server 2012 R2.</p> <p>Les agences régionales sont reliées à l'agence centrale grâce au WAN de l'entreprise, avec de la fibre optique. De plus, nous utilisons un firewall pour sécuriser les communications entre l'agence régionale et la centrale.</p> <p>Un firewall est mis en place sur la connexion en fibre vers l'extérieur du réseau de l'entreprise pour en assurer la sécurité, ce firewall est doublé en cas de panne du premier.</p>
<b>Qualité de service</b>	Haute disponibilité (> 90%) seulement pendant les heures d'ouverture de la bourse, que l'on étend aux périodes de pré-ouverture/pré-clôture soit de 7h30 à 17h35. Les samedis et dimanches ainsi que les nuits permettent donc d'exécuter les batchs et les mises à jour.
<b>Dimensionnement</b>	Deux serveurs sont à prévoir, en effet si une panne survient il est impératif qu'un second serveur soit déjà configuré et prêt à être utilisé. Chaque agence régionale possèdera ses propres équipements.
<b>Architecture d'exploitation</b>	Des tests de reprises sont conduits tous les mois afin de s'assurer de la bonne connaissance et de la maîtrise du processus de reprise. Une sauvegarde des données est effectuée toutes les semaines.
<b>Coût</b>	<p>Dell PowerEdge T110 II : 2000€ (HT/unitaire)</p> <p>Microsoft Windows Server 2012 R2 : inclus</p> <p>Firewall : Cisco Catalyst 6500 : 1000€</p>

## 8. Bloc applicatif « Agence locale »

<b>Architecture physique</b>	Matériel utilisé : Les agents des agences locales vont utiliser des PC Dell OptiFlex 7010 équipé du système d'exploitation Microsoft Windows Seven. Les agences locales sont reliés à l'agence régionale dont ils dépendent grâce au WAN de l'entreprise, avec de la fibre optique. De plus, nous utilisons un firewall pour sécuriser les communications entre l'agence locale et la régionale. Un firewall est mis en place sur la connexion en fibre vers l'extérieur du réseau de l'entreprise pour en assurer la sécurité, ce firewall est doublé en cas de panne du premier.
<b>Qualité de service</b>	Disponibilité de 80 %, disponible de 10h à 17h et fermé le lundi. Les agences locales n'effectuent pas de batch car ils ne disposent pas de logiciels nécessitant de mises à jour. L'application web qu'ils utilisent est mise à jour indépendamment.
<b>Dimensionnement</b>	10 postes utilisateurs par agence avec un system Windows 7, en cas de panne, les opérateurs font une demande support.
<b>Architecture d'exploitation</b>	N/A
<b>Coût</b>	Dell OptiFlex 7010 : 814€ (HT/unitaire)