1-Définir me modèle d’architectures en couche et en tiers

2-Préconiser des motifs de conception pour les couches

3-Préconiser les librairies, composants

**Stock information :**

Le logiciel implémenté va servir de parseur des données du marché. Il devra pouvoir rendre persistante 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++.

**Couches :**

|  |  |
| --- | --- |
| **Présentation** | N/A : aucune interaction avec l’utilisateur |
| **Contrôleur** | Doit contrôler l’appel de services, et doit contrôlé que le carnet d’ordre présent dans la base de données est à jours. |
| **Services** | Web services REST   * Récupérer un carnet d’ordre * Passer un ordre |
| **Domaine** | **Objets métiers :**   * Carnet d’ordres (liste d’ordres)   Utilisation de la libraire STL   * Ordre   **Fourniture des moyens d’accès à l’information :**  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. |

**Tiers :**

Modèle 2-tiers.

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

1 serveur Données : Oracle

**Transaction sytem :**

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

**Couches :**

|  |  |
| --- | --- |
| **Présentation** | N/A : aucune interaction avec l’utilisateur |
| **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) |
| **Domaine** | **Objets métiers :**   * Carnet d’ordres (liste d’ordres)   Utilisation de la libraire STL   * Ordre * Client   **Fourniture des moyens d’accès à l’information :**  Les accès à l’information son géré par le bloc Account Information |
| **Persistance** | La persistance des données est gérée par le bloc Account Information |

**Tiers :**

Modèle 1-tiers.

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

**Account information :**

Le logiciel implémenté va servir de persisté 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++.

**Couches :**

|  |  |
| --- | --- |
| **Présentation** | N/A : aucune interaction avec l’utilisateur |
| **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   * Créer un client * Supprimer un client * Modifier un client * Ajouter un ordre * Supprimer un ordre * Modifier un ordre |
| **Domaine** | **Objets métiers :**   * Carnet d’ordres (liste d’ordres)   Utilisation de la libraire STL   * Ordre * Client   **Fourniture des moyens d’accès à l’information :**  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. |

**Tiers :**

Modèle 1-tiers.

1 serveur Données : Oracle SPARC T4-1

**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.

**Couches :**

|  |  |
| --- | --- |
| **Présentation** | L’application fournit un client léger implémenté en ASP .net. Cette application dédiée aux clients individuels fournit les services de passage d’ordre. |
| **Contrôleur** | Doit contrôler l’appel des services, doit contrôler l’authentification des clients, la cinématique des écrans et les sessions utilisateur. |
| **Services** | Web services SOAP   * S’authentifier * Récupérer carnet d’ordre * Passer un ordre * Annuler un ordre * Modifier un ordre * Créer un compte utilisateur * Modifier un compte utilisateur |
| **Domaine** | **Objets métiers :**   * Carnet d’ordres (liste d’ordres)   Utilisation de la libraire STL   * Ordre * Client   **Fourniture des moyens d’accès à l’information :**  Les accès à l’information son géré par le bloc Account Information |
| **Persistance** | La persistance des données est gérée par le bloc Account Information |

**Tiers :**

Modèle 3-tiers.

1 serveur Client : Client (ASP .Net)

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

1 Serveur Données : Données (Account Information)

**Application Trader :**

Le logiciel implémenté est dédié aux traders de l’entreprise. Il doit permettre de recevoir les ordres et carnets d’ordres du marché en quasi temps réel, et doit pouvoir passer des ordres très simplement et rapidement.

**Couches :**

|  |  |
| --- | --- |
| **Présentation** | 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. |
| **Contrôleur** | Doit contrôler l’appel de services |
| **Services** | Web services REST   * Récupérer un carnet d’ordre * Passer un ordre |
| **Domaine** | **Objets métiers :**   * Carnet d’ordres (liste d’ordres)   Utilisation de la libraire STL   * Ordre   **Fourniture des moyens d’accès à l’information :**  Les accès à l’information son géré par le bloc Stock Information |
| **Persistance** | La persistance des données est gérée par le bloc Stock Information |

**Tiers :**

Modèle 1-tiers.

1 serveur Application :

**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.

**Couches :**

|  |  |
| --- | --- |
| **Présentation** | 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. |
| **Contrôleur** | Doit contrôler l’appel de services |
| **Services** | Web services SOAP C#   * Récupérer un carnet d’ordre * Passer un ordre |
| **Domaine** | **Objets métiers :**   * Carnet d’ordres (liste d’ordres)   Utilisation de la libraire STL   * Ordre   **Fourniture des moyens d’accès à l’information :**  Les accès à l’information son géré par le bloc Stock Information |
| **Persistance** | La persistance des données est gérée par le bloc Stock Information |

**Tiers :**

Modèle 1-tiers.

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

**Agence Régional :**

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 utilisé par les agents de l’entreprise.

**Couches :**

|  |  |
| --- | --- |
| **Présentation** | 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. |
| **Contrôleur** | Doit contrôler l’appel de services |
| **Services** | Web services SOAP C#   * Récupérer un carnet d’ordre * Passer un ordre |
| **Domaine** | **Objets métiers :**   * Carnet d’ordres (liste d’ordres)   Utilisation de la libraire STL   * Ordre   **Fourniture des moyens d’accès à l’information :**  Les accès à l’information son géré par le bloc Stock Information |
| **Persistance** | La persistance des données est gérée par le bloc Stock Information |

**Tiers :**

Modèle 3-tiers.

1 serveur Client : Client léger (ASP .Net)

1 Serveur Logique : Web Services (Web Application)

1 Serveur Données : Données (Account Information)

**Agence Local :**

Le bloc applicatif agence Local n’a pas de couche logicielle dédiée. Elle utilise la couche logicielle du bloc applicatif Agence Régional. Le personnelles de l’entreprise utilisera le client léger de l’agence Régional à partir d’un poste de travail.

**Gros client Application:**

Le logiciel va implémenter tous les web services et l’application web destiné aux gros clients. Les web services seront utilisés par l’application web dédié aux gros clients ainsi que par les applications externe tels que des applications mobiles dédiés à ces mêmes clients.

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

**Couches :**

|  |  |
| --- | --- |
| **Présentation** | L’application fournit un client léger implémenté en ASP .net. Cette application dédiée aux gros clients individuels fournit les services de passage d’ordre. |
| **Contrôleur** | Doit contrôler l’appel des services, doit contrôler l’authentification des clients, la cinématique des écrans et les sessions utilisateur. |
| **Services** | Web services REST   * S’authentifier * Récupérer carnet d’ordre * Passer un ordre * Annuler un ordre * Modifier un ordre * Créer un compte utilisateur * Modifier un compte utilisateur |
| **Domaine** | **Objets métiers :**   * Carnet d’ordres (liste d’ordres)   Utilisation de la libraire STL   * Ordre * Client   **Fourniture des moyens d’accès à l’information :**  Les accès à l’information son géré par le bloc Account Information |
| **Persistance** | La persistance des données est gérée par le bloc Account Information |

**Tiers :**

Modèle 2-tiers.

1 serveur Client + Logique : Client + Web Services (C# / ASP .Net)

1 Serveur Données : Données (Account Information)