Logiciel de gestion des ressources numériques

Architecture du Logiciel

FANTASTIC FIVE Team

20/05/2015

Historique des révisions

Date	Version	Description	Auteur
20/05/2015	<1.1>	Analyser l'architecture du logiciel.	• Bouazzaoui

Table des matières

Architecture du Logiciel

I. INTRODUCTION

Ce document est une description de l'architecture de l'application, il vise a donné une idée claire sur l'architecture globale de l'application (les couches, les relations entre elles).

II. OBJECTIF DU LOGICIEL

1. Contexte

Cette application résout le problème pos é par le client et qui consiste à effectuer la recherche de l'emplacement d'une ressource (document, dossier, vidéo, formation,..)Qui se trouve sur différents périphériques de stockage (machine, disque dur, USB,..) ainsi que la description de ces ressources à partir d'une seul machine via une interface graphique.

2. Besoins fonctionnels

Les besoins fonctionnels de l'application sont :

- Consulter l'emplacement des ressources disponible.
- Rechercher les ressources selon des paramètres.
- Consulter la liste des ressources.
- Consulter la liste des documents communs et non communs.
- Comparer les versions.
- Gérer les machines et les supports de stockage.

3. Besoins non fonctionnels

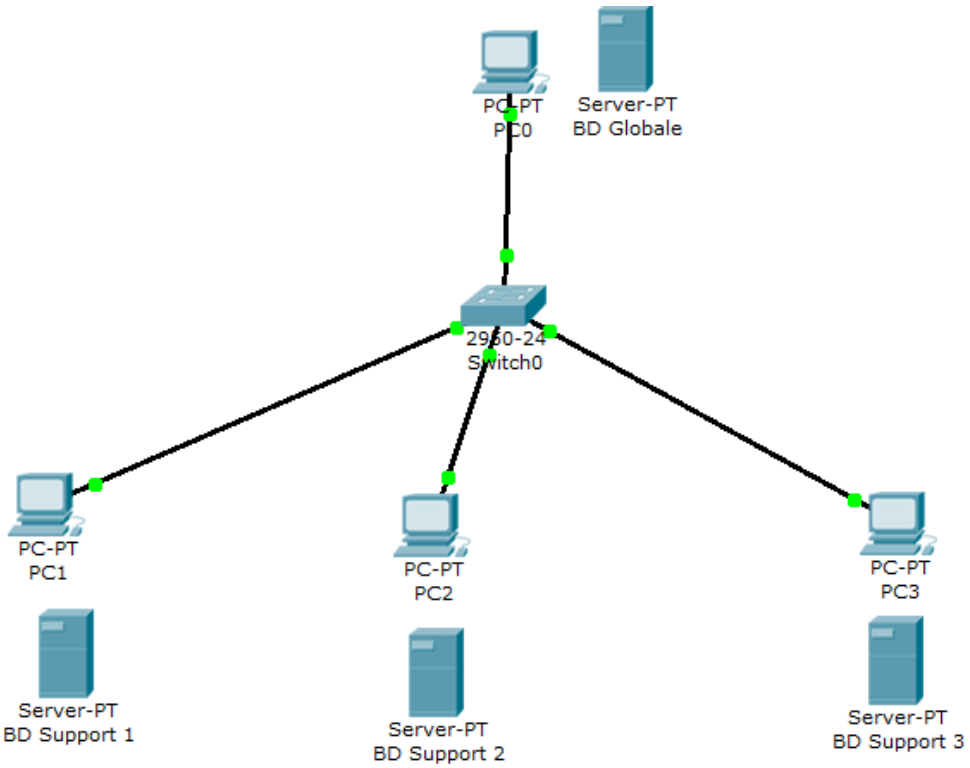
Des qualités et des contraintes importantes a prendre en compte lors de l'élaboration de l'architecture de l'application :

- ✓ Fonctionnalité
- ✓ Utilisabilité
- ✓ Fiabilité
- ✓ Disponibilité
- ✓ Temps moyen entre les pannes
- ✓ Temps moyen de réparation
- ✓ Performance
- ✓ Temps de réponse
- ✓ Capacité

III. STRUCTURE

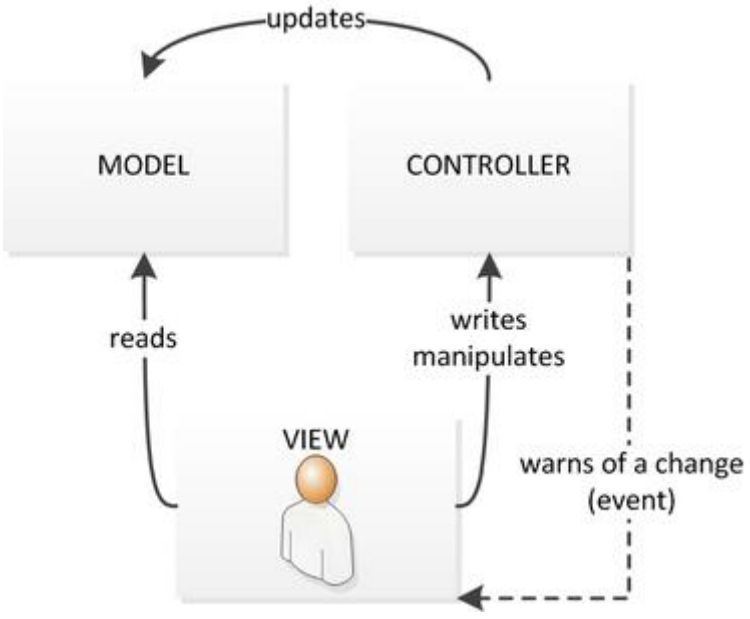
1. Vue des couches

L'architecture de notre système décrite dans l'image ci-dessous se base sur le principe client-serveur. En effet il y'aura une machine centrale représentant le serveur contenant la base de donnée globale qui sera mise à jour chaque fois qu'une base de donnée locale déployée sur une machine cliente sera mise à jour, la synchronisation s'effectuera via l'application qu'on va réaliser et qui sera implémenté sur toutes les machines .Les machines seront connectées via un réseau TCP/IP.



2. Sous-systèmes et paquetages

Notre code respectera le pattern MVC qui découpe le code en trois package comme décrit ci-dessous :



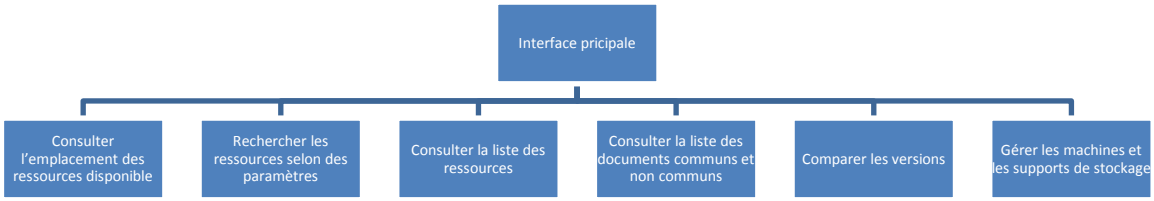
Modèle : Le modèle représente le cœur (algorithmique) de l'application : traitements des données, interactions avec la base de données, etc.

Vue : C'avec quoi l'utilisateur interagit se nomme précisément la vue.

Contrôleur : Le contrôleur prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser.

3. Interfaces

Notre application aura l'arborescence des interfaces suivante :



IV. COMPORTEMENT

1. Réalisation des cas d'utilisation

On fera la description du cas d'utilisation le plus général et le plus usuel et qui consiste à ajouter des fichiers a une machine. Cela implique une mise à jour de la base de données locale de la machine. Celle-ci doit être synchronisée avec la base de données globale implémentées sur la machine centrale .La synchronisation se fera comme suit : la base de données locale va faire un update des ressources de la base de données globale et qui ont comme emplacement la machine concernées et puis la base globale va écraser la base locale puis que c'est elle qui contient la dernière version des emplacements des ressources.

Les autres cas d'utilisation CRUD (Create, Read, Update, Delete) d'une machine ou périphérique de stockage n'a rien de spécial pour être décrite.

2. Mécanismes

✓ *Technologie*

Plusieurs possibilités de développement, ci-dessous : le listing des avantages et inconvénients de chacun :

Développement	Avantages	Inconvénients
Java	*Connaissance de la technologie*	*Nécessite de développer TOUS les scénarios de l'application*
Bases de données MySQL	*Connaissance de la technologie*	

✓ *Qualité de l'architecture*

L'architecture de notre application va nous permettre de donner des résultats compétitifs en terme de temps de recherche par rapport à nos concurrents dans le marché en l'occurrence TOTAL COMMENDER et l'outil de recherche intégré avec MS Windows.

Cette architecture va nous permettre de trouver des résultats fiables si les entités de notre système sont connectées et les bases de données sont bien synchronisées.