



Projet RMIAGE : Cahier des Charges

Badre Baba, Geoffroy Carrier, J-Ch Saad-Dupuy, Mickael Scheer

24 avril 2009

1 Présentation générale

1.1 Projet

1.1.1 Finalités

La réalisation qu'effectuera notre équipe est un *framework* accélérant le développement d'architectures de « réseaux sociaux » basés sur des clients lourds.

1.2 Contexte

Le développement des réseaux et des moyens de communication offrent désormais beaucoup de facilité pour communiquer et échanger du contenu de nature très diverse et ceci de manière aussi très variée. Quelque soit le contenu et la manière d'organiser ces espaces de communication où l'on partage, échange des informations, leurs objectifs sont très proches. On le voit très nettement dans l'avènement des « réseaux sociaux » qui les déclinent sous des formes très variées.

Notre client souhaite se placer sur ce créneau des logiciels d'échange et pour cela envisage de faire développer un « framework » suffisamment adaptable (système d'échange générique) pour répondre rapidement à des diffuseurs de contenu qui voudraient proposer à leurs propres clients une zone d'échange bien spécifique. Ces clients, très exigeants en ce qui concerne l'ergonomie des logiciels qu'ils proposent nous imposent des critères de réutilisabilité, d'interopérabilité et de généricité qui seront respectées lors de la réalisation.

1.2.1 Situation du projet par rapport aux autres projets de l'entreprise

Bien que notre équipe soit compétente et performante en développement, notamment dans les technologies Java, il s'agit ici de notre première réalisation commerciale.

1.2.2 Études effectuées

L'équipe a développé au cours des années précédentes une connaissance approfondie du marché des réseaux sociaux à travers une veille technologique. Dans ce cadre, les produits facebook, MySpace, twitter, identi.ca, LinkedIn, viadeo, youtube, flickr, meetic ont été testées en profondeur.

1.2.3 Nature des prestations demandées

1.2.4 Parties concernées par le déroulement du projet et ses résultats

Durant toute la durée du projet, nous serons en contact avec le maître d'œuvre, M. François Puitg.

Les utilisateurs finaux seront les clients exploitant des produits développés par notre client, diffuseurs de contenu, auprès de qui nous n'interviendrons pas.

1.3 Énoncé du besoin

Les utilisateurs finaux, diffuseurs de contenu, souhaitent proposer à leurs propres clients des zones d'échange bien spécifiques et nécessitent une architecture logicielle réutilisable leur mettant à disposition les fonctionnalités de bases pouvant être spécialisées suivant leurs besoins.

1.4 Environnement du produit recherché

Le *framework* doit être réalisé en Java, et utiliser la technologie RMI (Remote Method Invocation) pour la communication entre client lourd et serveur applicatif.

1.4.1 Listes exhaustives des éléments

Le produit livré se composera de trois parties :

1. Le *framework* ;
2. un programme témoin d'application serveur ;
3. un programme témoin d'application cliente.

1.4.2 Caractéristiques

1. *Framework*

Le *framework* permettra une présentation hiérarchique des données côté client, et reposera sur la technologie de persistance JPA permettant de stocker les données sur un serveur SQL, assurant leur cohérence, la fiabilité de stockage et la rapidité des recherches en leur sein. Il permettra l'ajout, la suppression, la modification de celles-ci à travers une couche de sécurité adaptable aux besoins métier.

Il fournira une application lourde adaptable aux besoins et dont composants éléments métier pourront être fournis à la volée par le serveur applicatif.

Les arborescences d'informations des utilisateurs pourront partager des branches grâce à une gestion en graphe des données.

Des modules de gestion de contacts et de discussion seront fournis, proposant en standard des fonctionnalités fréquemment proposées sur les réseaux sociaux.

2. Application témoin serveur

L'application témoin serveur sera un simple *package* configuré pour utiliser le serveur SQL embarqué dans la machine virtuelle Java de Sun et ne

nécessitant aucune configuration, auquel sera adjoint un système de forums (fils de discussion publics), et un modèle de sécurité minimal.

3. Application témoin cliente

Il s'agira de l'application déployable directement par les clients, et capable *via* la diffusion des éléments graphiques métier par le serveur de s'adapter aux différents réseaux proposés.

Elle sera néanmoins adaptable à volonté par le client au cas par cas (ajouts de fonctionnalités, d'une documentation ad hoc, thèmes visuels, retrait du choix du serveur, etc.).

Un prototype de l'interface permettant d'en percevoir la conception générale est proposé en annexe B.

2 Expression fonctionnelle du besoin

2.1 Fonctions de service et contrainte

2.1.1 Fonctions de service principales

Le framework doit permettre de créer et d'utiliser différents type d'objets génériques, permettant aux application l'utilisant de créer :

- des utilisateurs
- des groupes d'utilisateurs
- des structures pour des échanges de fichiers
- des structures pour des échanges de messages

2.1.2 Fonctions de service complémentaires

Deux programmes de démonstration exploitant les fonctionnalités du framework doivent également être fournis :

- un programme serveur basé sur notre produit proposant un système de forums ;
- un programme client, exploitant les ressources fournies par le serveur.

De plus, nous fournirons une batterie de tests unitaires garantissant la qualité de chaque fonctionnalité du *framework*.

2.1.3 Contraintes

1. Développement

Le framework sera aussi indépendant que possible, notamment de tout éditeur de serveur SQL grâce à la technologie Hibernate.

2. Environnement

Notre framework se composera d'une librairie, utilisée par les applications témoins livrées.

L'utilisation de la technologie Java assure une complète indépendance du ou des systèmes d'exploitations hôtes.

Les applications serveur et cliente pourront être lancée depuis le même poste, en local, ou sur des machines distantes.

Les machines hôtes et serveur devront chacune disposer d'une Java Virtual Machine (JVM) afin de pouvoir exécuter ces applications.

2.2 Critères d’appréciation

Les principaux critères d’appréciation seront :

- du côté *framework* :
 - son indépendance ;
 - sa qualité ;
 - sa flexibilité ;
- du côté applications témoin :
 - leur pertinence vis-à-vis des fonctionnalités principales du *framework* ;
 - la facilité de leur utilisation pour des tests.

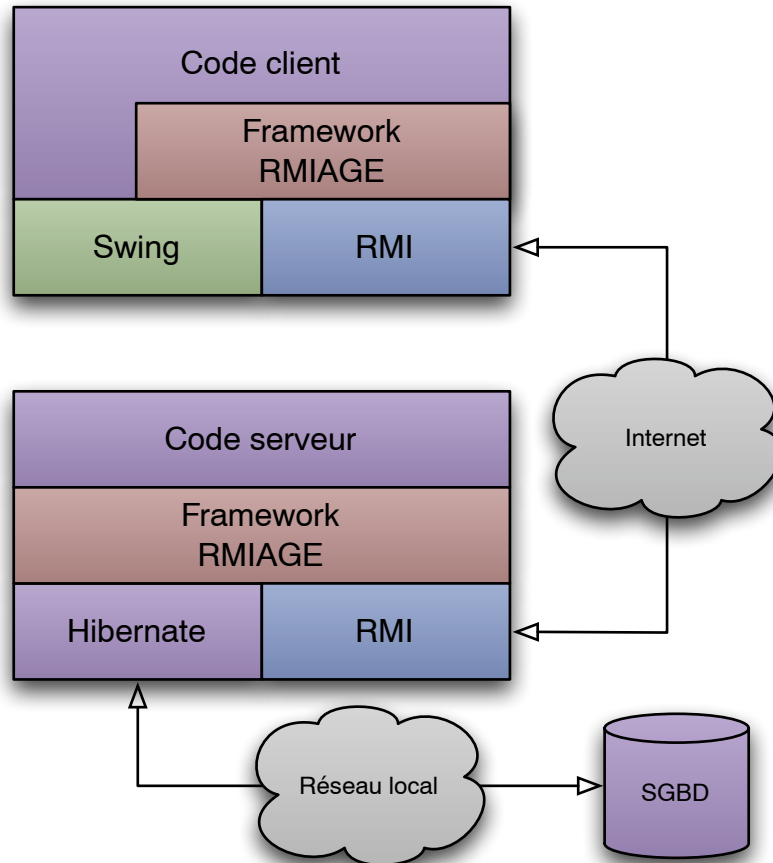
3 Cadre de réponse

3.0.1 Outils d’installation, de maintenance

Le produit sera livré sous forme d’archives JAR contenant la librairie et les programmes témoins.

Les instructions concernant la configuration d’Hibernate seront également fournies.

A Architecture générale de l'application



B Interface de l'application témoin Cliente

C Estimation de l'avancée du projet

C.1 Diagramme de Gantt

N°	Liste des taches	Jours	Dépendance
1	Analyse des besoins	1	
2	Choix de l'architecture logicielle	1	
3	Réalisation de l'audit N°1	1	1,2
4	Documentation	3	3
5	Réalisation du cahier des charges	2	3
6	Réalisation diagramme UML	2	3
7	Réalisation de l'audit N°2	1	5
8	Spécification des interfaces	3	5
9	Réalisation de l'architecture logicielle	4	5,6
10	Réalisation de l'audit N°3	1	6,7,8
11	Réalisation du cas particulier	3	9,10
12	Réalisation des tests	1	11