



# 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 à effectuer par notre équipe porte sur le développement logiciel d'une bibliothèque (framework) permettant des échanges de données au sein de structures logicielles développées par le client, de type "réseau social"

### 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 développer un « framework » suffisamment adaptable (système d'échange générique) pour répondre rapidement à des diffuseurs de contenus 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 Parties concernées par le déroulement du projet et ses résultats

Durant toute la durée du projet, nous serons en contact avec le représentant du comité de pilotage, M. François Puitg.

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

### 1.3 Enoncé du besoin

Les utilisateurs finaux, diffuseurs de contenus, souhaitent proposer à leurs propres clients des zones d'échanges 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, en utilisant la technologie Remote Method Invocation (RMI).

#### 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 organisation hiérarchique des données. Il mettra à disposition des méthodes permettant des fonctions d'ajout-suppression-modification de celles-ci. Il permettra également de générer des objets d'interfaces que l'application serveur pourra transmettre à l'application cliente. L'aspect social sera réalisé en intégrant au framework la possibilité à chaque utilisateur de créer des groupes de contacts et d'échanger des messages avec plusieurs personnes.

##### 2. Application témoin serveur

L'application témoin serveur sera basée sur notre framework. Il intégrera un protocole d'échange avec les applications clientes, basées sur RMI. Il implémentera des spécialisations des objets génériques du framework, permettant une gestion d'utilisateurs, par groupes et permissions. Il assurera de même la gestion de la persistance des données.

##### 3. Application témoin cliente

L'application témoin cliente communiquera avec l'application serveur. Il s'agira d'une application générique, utilisant le protocole RMI pour obtenir les objets d'interface graphique permettant l'affichage et la saisie des données.

L'interface attendue pour cette application témoin est donnée 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 strucures pour des échanges de fichiers
- des strucures 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 ainsi que la technologie RMI, mettant à disposition des ressources ;
- un programme client, exploitant les ressources fournies par le serveur.

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

#### **2.1.3 Contraintes**

##### **1. Développement**

Le framework doit être au maximum indépendant, notemment au niveau de l'accès au données. Le client doit avoir la possibilité d'utiliser le système de gestion de base de données relationnelle (SGBDR) de son choix.

##### **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 pourrons être lancée depuis le même poste, en local, ou sur des machines distantes.

Les machines hôtes et serveur devrons chacune disposer d'une Java Virtual Machine (JVM) afin de pouvoir executer 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
  - leur exhaustivité par rapport aux fonctionnalités offertes
  - leur facilité de mise en place

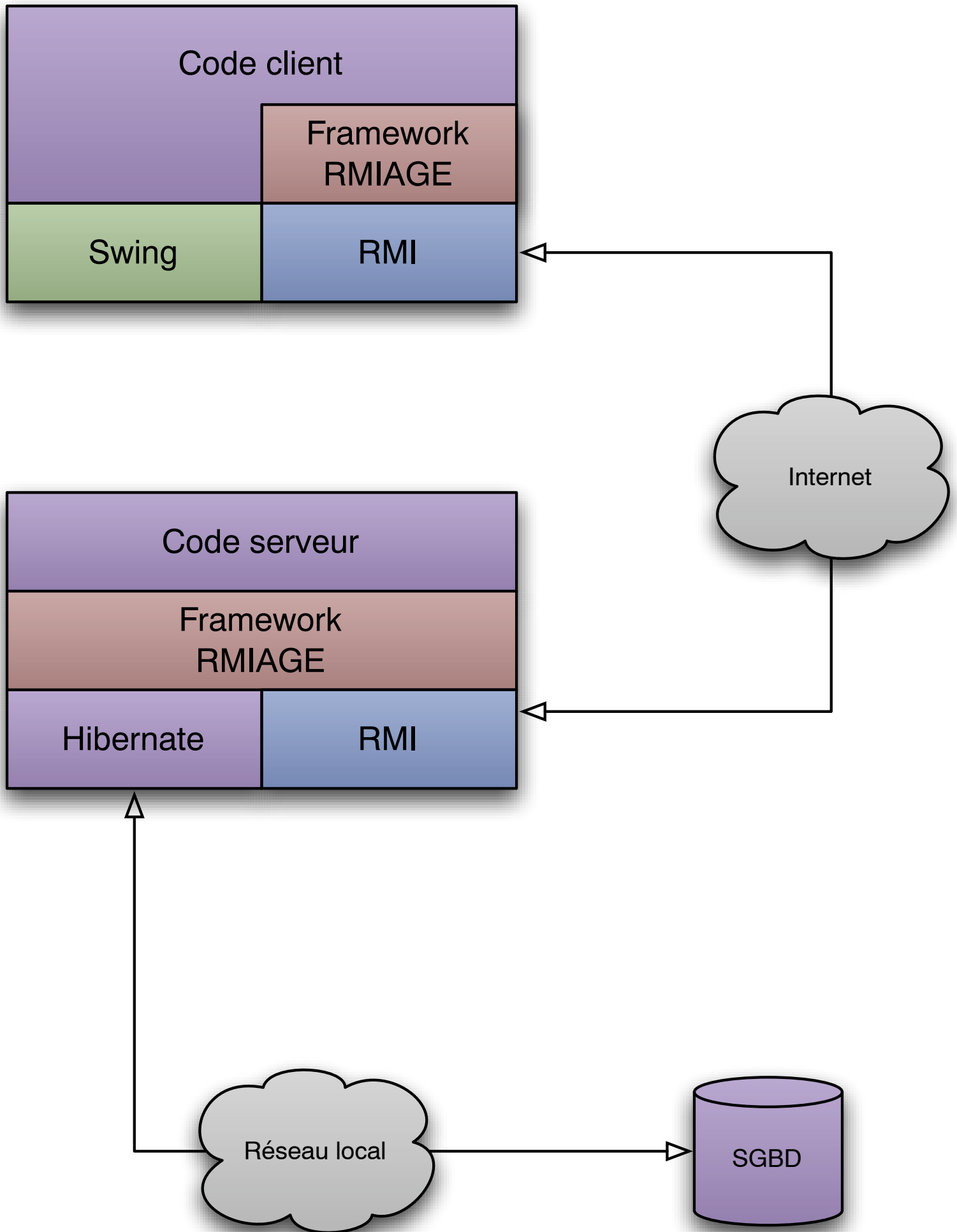
## **3 Cadre de réponse**

### **3.0.1 Outils d'installation, de maintenance**

Le produit sera livré sous forme d'archives contenant les executables de la librairie et des programmes témoins (Jar).

Les instructions concernant la configuration du système de persistance des données adoptée 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         |