

# Dossier technique

---

“ *KICQA* ”  
By Groupe Lagalère<sup>™</sup>

Projet L3 MIAGE  
Groupe 1  
Amin Carine - Blondel Quentin - Charron Antoine -  
Perrin Kyllian - Thierno Ibrahima Sory Diallo

## **SOMMAIRE**

<b>Introduction</b>	<b>3</b>
<b>1. Description générale du produit</b>	<b>4</b>
1.1. Description du projet	4
1.2. Description de la solution	4
1.3. Contraintes et choix techniques	4
<b>2. Description technique du produit</b>	<b>5</b>
2.1. Principe de fonctionnement	5
2.2. Architecture du framework	5
2.3. Installation de la solution	5
2.4. Extension du framework	6

## Introduction

La solution KICQA By Groupe Lagalère™ c'est avant tout la simplicité permettant de réaliser et déployer rapidement une application d'échanges numériques. Basé sur un framework entièrement modulable, notre solution propose un chat multi-utilisateurs, une gestion des documents et un calendrier.

Le framework peut être enrichi de nouvelles fonctionnalités par l'intégration de nouveaux modules ou la modification des modules existants. Le développeur pourra alors s'appuyer sur la documentation des méthodes existantes pour réaliser ses propres fonctionnalités.

## **1. Description générale du produit**

### **1.1. Description du projet**

Ce projet répond à l'appel d'offre lancé par la société "M2I" visant à la réalisation et la mise en oeuvre d'un framework d'échange numérique accompagné d'une application témoin.

Ce projet a été réalisé par notre société, spécialisée dans la réalisation de solutions informatiques telles que des logiciels, progiciels, frameworks... et qui a donc remporté l'appel d'offres de la société "M2I". Cette solution a donc été développée par notre équipe composée de 4 développeurs et un chef de projet/développeur.

### **1.2. Description de la solution**

La solution KICQA proposée par notre entreprise se compose de deux lots : à la fois un framework, ainsi qu'une application témoin permettant d'illustrer les fonctionnalités du framework. Ce dernier permet la mise en oeuvre d'une plateforme d'échange et de communication entre différents utilisateurs, proposant les fonctionnalités suivantes :

- Chat : communication par envoi de messages entre différents utilisateurs,
- Calendrier : programmation d'événements à partager avec d'autres utilisateurs,
- GED (gestion électronique de documents) : partage de documents avec d'autres utilisateurs.

Le framework se présente sous la forme d'une application client/serveur en JAVA et utilisant la technologie RMI pour la communication entre le client et le serveur.

### **1.3. Contraintes et choix techniques**

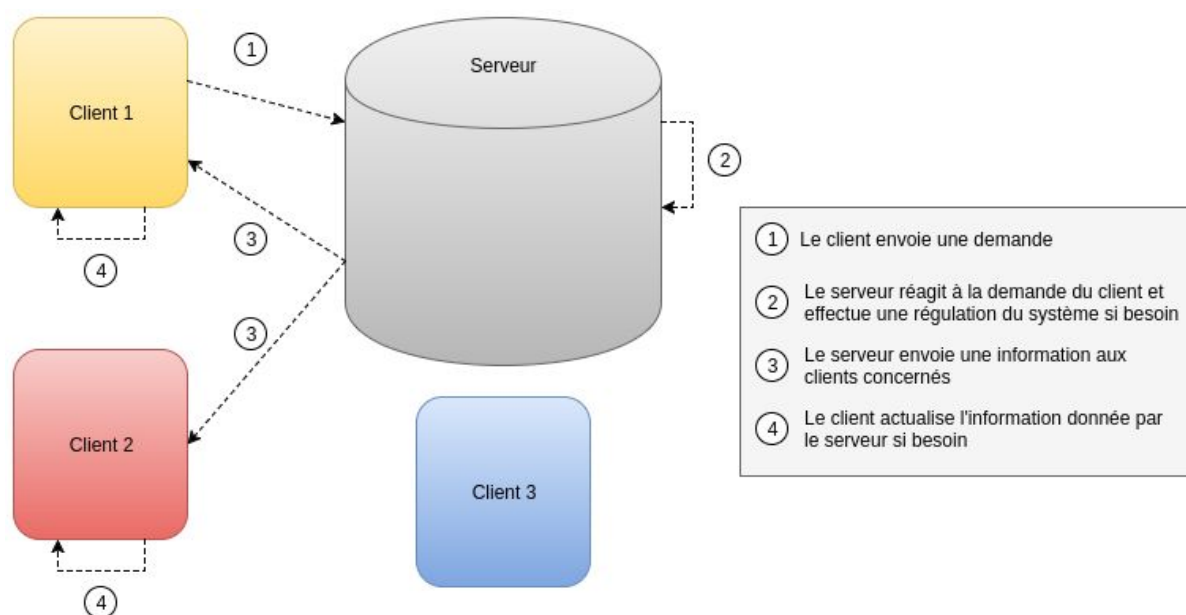
Le framework se devait de rester un maximum modulable afin que l'entreprise "M2I" puisse l'utiliser dans la conception de différentes applications d'échanges numériques. Le framework devait également être réalisé en Java.

En plus de ces contraintes techniques, nous avons été amené à faire différents choix sur les technologies employées pour la réalisation de notre solution. Le framework utilise ainsi le SQL puisqu'il s'appuie sur une base de données, notamment pour la gestion des utilisateurs, des documents et du calendrier. Nous avons également utilisé le XML qui permet d'enregistrer les profils des utilisateurs et d'archiver les conversations pour la partie Chat du framework. De plus le framework s'appuie sur la technologie RMI pour la communication entre la partie client et la partie serveur. Il a été fait le choix aussi de ne pas contraindre le futur développeur du framework en ne retournant pas une chaîne de

caractère mais plutôt un tableau associatif des valeurs proposées par le framework. Le développeur pourra alors moduler l'information comme il le souhaite.

## 2. Description technique du produit

### 2.1. Principe de fonctionnement



### 2.2. Architecture du framework

Le framework se décompose en 4 fonctionnalités : Calendrier, Chat, GED et gestion des utilisateurs. Chaque partie possède deux interfaces qui définissent les méthodes côté client et côté serveur ainsi qu'une partie correspondant aux fonctionnalités nécessaires : dans le côté Client, nous retrouvons toutes les fonctions que la future application aurait besoin pour interagir avec le framework et dans le serveur toutes les fonctions qui servent à la gestion de ces données. Normalement, aucun appel de serveur ne doit être effectué depuis l'application afin d'éviter tout risque de sécurité, tout doit passer par la partie client. Les interfaces permettent ici de définir quelles fonctions sont disponibles depuis l'autre utilisateur afin de pouvoir garder une certaine sécurité définie par le développeur.

### 2.3. Installation de la solution

Comme dit précédemment, la solution se présente sous la forme d'une application client/serveur. Pour le déploiement de la solution, il faut ainsi une machine jouant le rôle de serveur sur lequel sera installé la partie serveur. La partie client quant à elle devra être installée sur la machine de chaque utilisateur.

De plus comme le framework s'appuie sur une base de données, il faut également mettre en place un serveur Oracle et créer la base de données avec les scripts fournis dans la solution.

## **2.4. Extension du framework**

Notre framework est adaptable et peut être spécialisé en fonction de l'application finale que l'on souhaite mettre en oeuvre. Différentes possibilités avec différents niveaux de complexité ont été prévu afin d'étendre le framework :

- Adaptation de la base de données ou des modèles XML,
- Modification d'un module existant,
- Création d'un nouveau module.

À savoir que l'adaptabilité des champs XML et SQL s'appliquent sur tous les modules, y compris le noyau du framework.