**Jourdain Nicolas**

**PIETRZAK aurÉlien**

**BUCHARD Thomas**

**LEFEVRE Romain  
L3 Info - fa**

**Introduction aux nouvelles technologies des systemes d’information**

TP Micro Services avec Spring Boot et RabbitMQ

**INTRODUCTION**

Le projet consiste à développer une messagerie interne d’entreprise, qui permettra aux employés de discuter exclusivement intra-muros via un outil spécifiques et « sécurisé », mais également de permettre l’échange de fichiers de tous types entre employés via cette messagerie.

On va développer cette messagerie sur la base de micro services qui auront chacun en charge un élément de gestion de cette messagerie, vous développerez essentiellement les services Front & Back avec le Framework Spring Boot.

**CONSIGNES**

Le client de messagerie JavaFX, ou Swing ou web MVC permettra à chaque utilisateur de se connecter à la messagerie de l’entreprise via un couple (login/mdp) (C’est le Job du « Service Live Registrar ») cette connexion sera permise si et seulement si l’employé́ est connu de la base « User\_DB ». Chaque tentative de connexion ou réussite de connexion devra être logué dans cette base « User\_DB » dans une table « History ».

* La demande de connexion au « Service Live Registrar » se fera via un Web Service REST POST.
* Idem lors de la demande de déconnexion qui se fera au travers d’un WS-Rest de type DELETE.

Le « Service Live Registrar » permettra également de fournir l’ensemble des utilisateurs actuellement connecté à cette messagerie dès qu’un utilisateur se connecte (via l’envoie d’un message AMQP a l’ensemble des utilisateurs actuellement connectés, ie clients).

Le client permettra d’envoyer des messages texte et/ou des fichiers de tous type vers l’un des employés actuellement connecté. Ces messages circuleront au travers d’un bus AMQP (rabbitMQ), les messages seront transportés et délivrés jusqu’au destinataire si et seulement si le destinataire est encore en ligne à ce moment-là̀ (ce sera le Job du « Service IsAlive » de contrôler cela).

Nb : vous pouvez implémenter cette fonctionnalité́ en fin de projet s’il vous reste du temps, cette fonctionnalité étant en Option.

Tous les messages transportés devront être logués dans la base « Log DB Msg History », on y gardera le Qui, quoi, Quand, Ou (machine), tous les messages devront transiter via le bus AMQP sont arrivés vers le « Service Log Msgs » qui se chargera de loguer en base ces données.

Nb : On ne gardera pas l’intégralité́ du contenu des fichiers dans les logs, mais uniquement le nom et taille du fichier.

Dès réceptions d’un fichier dans un message, on stockera ce fichier dans le dossier courant du client et signalera à l’utilisateur qu’il a à disposition un fichier dont on donnera le nom et la taille.

Vous développerez également une interface JavaFX ou Swing ou (Web Mvc) Administrateur qui permettra d’enregistrer, supprimer et lister les utilisateurs de la messagerie (pas de droits particuliers), on affectera à chaque utilisateur un login=nom-prénom et un mot de passe.

Ces utilisateurs devront être créés dans la base « User\_DB », cette intégration sera prise en charge par le service « Service Primary Registrar ».

Enfin le client Administrateur pourra demander la génération d’un PDF ou autre format de votre choix contenant tous les logs d’une Date D à une date F (on se limitera à ces critères sauf s’il vous reste du temps). C’est le « Service log Editor » qui prendra en charge ce travail et fera un retour au client via une écriture d’un PDF ou autre type de doc vers un dossier.

**QUESTIONS**

1)  Quels sont les points faibles de cette architecture ?

2)  Proposez une architecture qui permet d’assurer un service continue sur la partie messagerie cliente, ici on ne prendra pas en compte la continuité́ de service de la partie administrative.

3)  RabbitMQ est un point névralgique de notre architecture, que peut-on faire pour que cela ne le soit plus, ou tout au moins permettre une solution de secours.

**CONCLUSION**