

## INF 2165 – Introduction aux systèmes distribués

### TP MOM 2 : Application bancaire avec OpenJMS

L'objectif du TP est de construire et d'exécuter une application distribuée Java simulant la gestion de comptes en banque en utilisant l'API JMS. Cette application fera usage des communications sur file de messages ainsi que des communications basées-contenu sur des topics.

L'application comporte les entités suivantes :

- Le gérant des comptes : il s'agit de l'entité conservant les données relatives à chaque compte et appliquant les opérations sur ce compte. Les renseignements relatifs à un compte sont : un numéro de compte (entier), une identité de client (chaîne de caractère), un historique des opérations et un solde. Les opérations possibles sont le dépôt et le retrait, caractérisés par un montant. La première opération sur un compte tiendra lieu d'ouverture du compte.
- Le client de compte : cette entité effectue des opérations sur un compte (il représente l'utilisateur au guichet) en envoyant des messages au gérant. Ces messages serviront à faire des dépôts et des retraits. Toutes les opérations seront datées. On pourra considérer plusieurs instances de clients correspondant chacune à un usager différent.
- Le département des placements (représentant un département de la banque ou un organisme extérieur spécialisé dans les placements). Cette entité surveille l'évolution du solde des comptes. Seuls les nouveaux comptes, ouverts dans l'année, l'intéressent (on assimilera la date d'ouverture d'un compte à la date de la première opération). Son objectif est de détecter que le solde dépasse un certain seuil à partir duquel il fera une proposition de placement financier à l'utilisateur (par un moyen de type courrier électronique qui ne sera pas géré par votre application).
- Le département des découverts : il est lui aussi intéressé par le solde, cette fois-ci pour tous les comptes. Pour chaque compte, il étudie l'évolution du solde et lorsque celui devient négatif, il fait retirer par le gérant une somme forfaitaire (e.g. 10 €) sur le compte.

## 1 Modélisation de l'application

Schématisez l'application sous la forme d'un diagramme où vous ferez clairement apparaître les processus mis en jeu ainsi que les *destinations* (au sens JMS, i.e. file et topic) qui supportent la communication entre ces processus.

Vous préciserez le mode de communication (*push* ou *pull*) pour chaque liaison.

## 2 Implantation

Le gérant, le département des prélèvements et le département des découverts pourront être implantés sous la forme de démons (processus ne terminant pas) et s'exécutant chacun dans son propre terminal. Ces processus afficheront les informations utiles à la compréhension de leur fonctionnement.

Le client de compte sera implanté par un programme dont l'exécution effectue une opération sur un compte puis se termine. On lui passera en paramètres les informations nécessaires à l'opération.

## 3 Exécution

Dans un premier temps, vous exécuterez tous les processus sur la même machine.

Dans un deuxième temps vous utiliserez trois machines des salles de TP : une pour le gérant, une pour les départements des prélèvements et des placements, et une troisième machine pour les clients.

## 4 API d'administration OpenJMS

Modifiez le code de l'application pour que l'ensemble des files et topics soient créés par le programme gérant, et non plus via l'interface graphique d'administration.

Puis modifiez l'application pour faire en sorte que les clients se répartissent entre onze guichets (numérotés de 0 à 10) lorsqu'ils communiquent avec le gérant. Le guichet numéro 10 est forcément présent mais plusieurs des autres guichets peuvent ne pas être créés. On supposera qu'un client doit s'adresser au guichet portant le numéro égal au dernier chiffre de son numéro de compte client, si ce guichet existe. Si ce guichet n'existe pas, le client s'adressera au guichet numéro 10.