

## Projet Informatique Répartie (RMI)

On se propose de simuler le scénario suivant : un cuisinier cherche des ingrédients pour préparer un repas au moindre coût.

Il va chercher les ingrédients un par un, en procédant ainsi :

- Pour un ingrédient donné, il interroge une liste de magasins, appelés Mag1 à MagN. Chacun de ces magasins se trouve sur une machine différente. Chaque magasin est géré par un processus appelé `StoreManager`, qui accède à un fichier `Mag(i)` qui contient la liste des ingrédients ainsi que leurs prix dans la ligne suivante,
- Le cuisinier est modélisé par un processus `StoreClient` qui s'exécute sur une machine quelconque. Ce processus va interroger, pour un ingrédient donné, une suite de processus du type `StoreManager` - à la fin de la consultation il affiche à l'écran le nom du site où se trouve le magasin proposant cet ingrédient au prix le plus bas. L'interrogation d'un processus du type `StoreManager` se fera par l'appel à distance de la méthode `getPrice` qui aura été rendue visible dans l'interface `Store` implémentée par `GerantMagasin`.

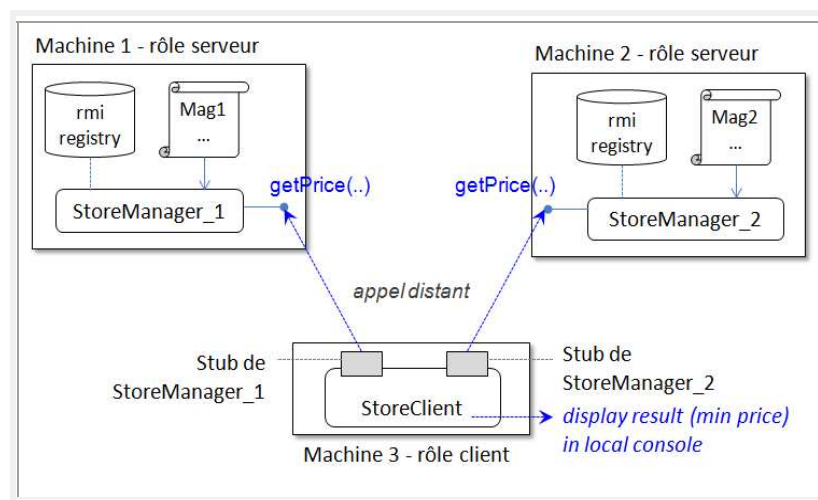


Figure 1: Schéma du mécanisme d'interrogation

Voici le contenu de l'interface `Store.java` :

```
/**
 * remote interface for Store services
 */
public interface Store extends java.rmi.Remote {
    /**
     * get the price of the given ingredient
```

```

    * note: this method can be called remotely
    * @param ingredient: the ingredient whose price should be returned
    * @return: the price of the ingredient
    * @throws java.rmi.RemoteException
    */
    float getPrice(String ingredient) throws java.rmi.RemoteException;
}

```

Remote signifie que la méthode `getPrice` des classes implantant cette interface `Store` peut être appelée depuis une JVM autre que la JVM locale.

`StoreManager.java` implémente cette interface :

```

public class StoreManager implements Store {
    ...
}

```

et `StoreClient.java` utilise la méthode publique `getPrice` pour interroger les magasins.

Et voici le contenu des fichiers `Mag1`, `Mag2` et `Mag3` qui contiennent les informations concernant l'ingrédient et son prix chacun dans une ligne :

Mag1	Mag2	Mag3
cornichons	cornichons	cornichons
10.0	20	50
safran	safran	safran
1000.0	500	6
sel	sel	sel
1.0	3	30
poivre	poivre	poivre
100.0	200	50

TAF :

Le menu de l'application du client est le suivant :

Sélectionner le numéro de l'ingrédient, puis cliquez sur « Entrée » :

[1] cornichons

[2] safran

[3] sel

[4] poivre

Voici un exemple de retour de résultat sur la console du client à la suite d'une recherche du prix le plus bas de l'ingrédient « safran » :

**Le prix le plus bas de l'ingrédient « safran » est de « 6 DT » et se trouve chez « Mag3 ».**