

## Travaux Pratiques

### FI BDCC S5 : Architectures Orientées Stream Processing

**Mohamed HAMMANE**

- **Introduction :**

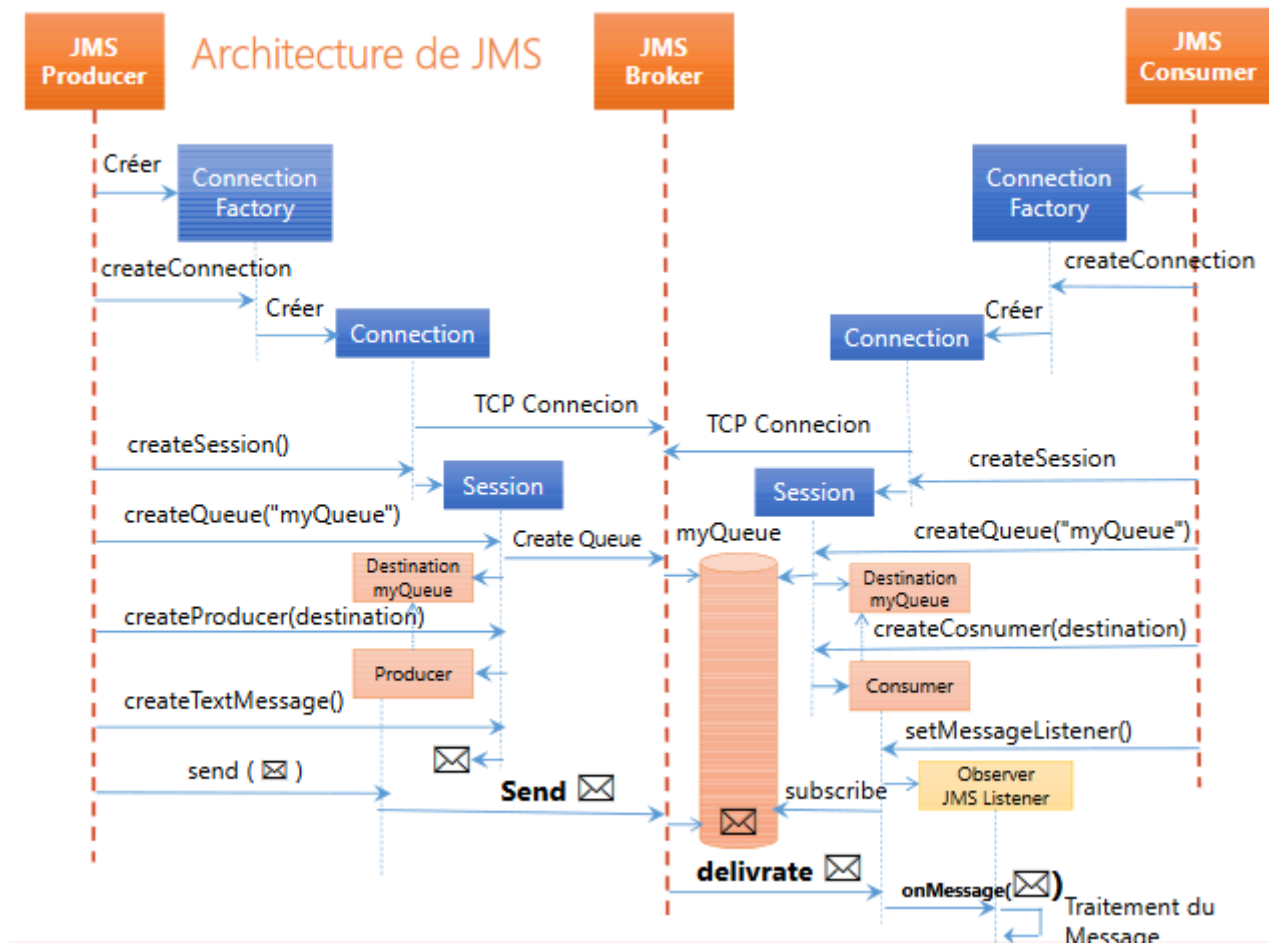
Stream processing est un paradigme de programmation, équivalent à dataflow programming et réactive programming, qui permet à des applications d'exploiter une forme limitée de parallélisme. Ces applications peuvent utiliser des unités de calcul, tel que les FPU sur un GPU ou FPGAs sans gestion explicite d'allocation, synchronisation ou communication entre eux.

Une des applications courantes du stream processing est le traitement analytique des données en temps réel. À l'aide des technologies de messaging comme Apache Kafka et des frameworks de stream processing (comme Apache Flink, Apache Spark ou Kafka Stream par exemple), il est possible de créer des pipelines de traitement de données complexes produisant des résultats en temps réel sur des technologies scalables et résilientes.

- **Enoncé :**

On veut créer une application basé sur ActiveMQ (Apache ActiveMQ est un courtier (broker) de messages open source écrit en Java avec un client complet de Java Message Service. Il fournit des fonctionnalités pour les entreprises, comme de simplifier et encourager la communication des applications par messages) qui utilise JMS (Java Message Service (JMS) permet d'envoyer et de recevoir des messages de manière asynchrone entre applications ou composants Java. JMS permet d'implémenter une architecture de type MOM (message oriented middleware). Un client peut également recevoir des messages de façon synchrone dans le mode de communication point à point. L'API JMS permet aux applications Java de s'interfacer avec des intergiciels (middleware) à messages ou MOM. Les MOM permettent des interactions entre composants applicatifs dans un cadre faiblement couplé, asynchrone et fiable), qui permet d'envoyer et de recevoir des messages (text et image) entre les différents producteurs qui ont déjà inscrit dans un topic.

## • Architecture :



## • Code source :

<https://github.com/Medomane/JMS-ActiveMQ> , avec une vidéo de démonstration.

## • Exécution :

Après le démarrage de ActiveMQ en utilisant la commande suivante

```
C:\Users\Wizardly\Projects\tools\apache-activemq-5.16.0\bin>activemq start
```

On a créé une application Maven nommé chat qui utilise JavaFX :

JMS Chat application

Code :  Host : localhost Port : 61616


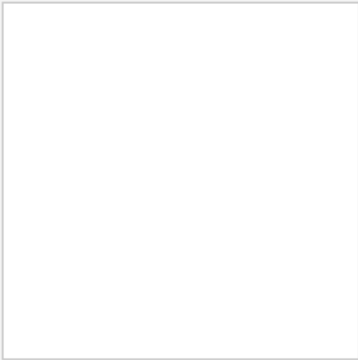
Message :

To :

Message :

Image :

Result



La première chose est la création de de consumer en fournissant un code, le host et le port.

Puis on a la zone du message à envoyer :

Message :

To :

Message :

Image :

On écrit le code du récepteur puis un message on peut choisir une image aussi puis on clique sur send.

On cette résultat par exemple.



- **Conclusion :**

L'utilisation du ActiveMQ basé sur JMS nous a aidé à résoudre beaucoup de problèmes. Mais est-ce qu'il y a une autre technologie plus avancée que ActiveMQ ?