DA *Distributed Algorithms*
May 2021

TD	MOM (Message-oriented middleware)

Objective of the session :
A first step with JMS and ActiveMQ.

1 Some MOM and JMS basics

Exercises

JMS (Java Message Service) is an API that provides the facility to create, send and read messages. It provides loosely coupled, reliable and asynchronous communication.

1. **Among the following applications, indicate whether the use of message-oriented middleware is appropriate, justify your decisions :**
   — **Video conferences.**
   — **Remote weather information query.**
   — **Messaging (e-mail, fax, voice mail).**
2. **What does it mean that the communication is asynchronous ?**
3. **Are JMS messages sent directly from an application that sends messages to another application that receives them ?**
4. **What are the two options in which a message can be published ?**

2 ActiveMQ and JMS

Exercises

Below is an example of how to use the Java JMS api with ActiveMQ running both topic-based and queue-based messaging.

1. **Be sure that you have installed :**
   — **Java SDK**
   — **Maven** [1]
2. **Download ActiveMQ 5 "Classic"** [2]**. Then, unzip it.**
3. **Open a terminal and navigate to the ActiveMQ directory. Start the server by executing** `bin/activemq console`
4. **Download the JMS-ActiveMQ-Example file from Moodle and unzip it.**
5. **Open three additional terminals (in addition to the terminal that ActiveMQ is running in). One terminal for a java application sending JMS messages to two others applications receiving them.**

1. http ://maven.apache.org/download.html
2. https ://activemq.apache.org/

6. **In one of the three terminals, run the `mvn install` command to build the example project.**
7. **Now we can start the JMS Consumers and Producer for Topic-based (`T`) and Queue-based (`Q`) Messaging in each previously opened terminal.**
   `java -cp target/jms-example-SNAPSHOT.jar example.Consumer Q` **or** `T`
   `java -cp target/jms-example-SNAPSHOT.jar example.Producer Q` **or** `T`
8. **If the `CLOSE` message is sent, this will close the connection to the consumers.**
   What happens when the CLOSE message is sent in the topic-based case ? What happens when the CLOSE message is sent in the queue-based case ? Why do you think this happens ?
9. **Modify/add message producers and message consumers so that a producer can send messages to a given consumer.**