**ActiveMQ Design Document**

1. **Technologies used:**

The tools that are to be used are:

1. ActiveMQ 5.15
2. Maven 3.5
3. **Features that are added along with the ActiveMQ implementation:**

* When assembling, we send the whole ASM image to the backend, and retrieving the response back as ASM image. When simulating, we again send the ASM image to the backend. Instead of sending the ASMImage between backend and frontend, we send a unique id associated with the ASMImage, so that when we click on the simulator, we can associate the id associated with the ASMImage, and can retrieve the ASMImage from the database.
* When simulating, we again send the ASM image to the backend.
* Instead of sending the ASMImage between backend and frontend, we send a unique id associated with the ASMImage, so that when we click on the simulator, we can associate the id associated with the ASMImage, and can retrieve the ASMImage from the database.
* The ActiveMQ connections is a two-way publisher subscriber model.
* The frontend is going to publish the message in a separate queue or topic, to which all the backend components are subscribed and vice versa.

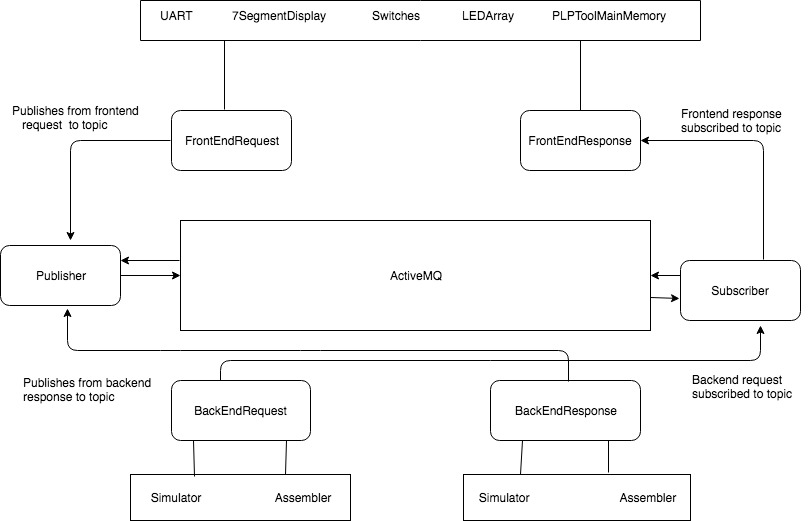
1. **Messaging Format Used:**

* We can use any type of format we would want to store in the ActiveMQ.
* However, it’s important for us to serialize and de serialize the message between the producer and consumer of the message.
* We can either use the Jackson API or the Gson API to do the conversion.

1. **Overall Design Diagram:**

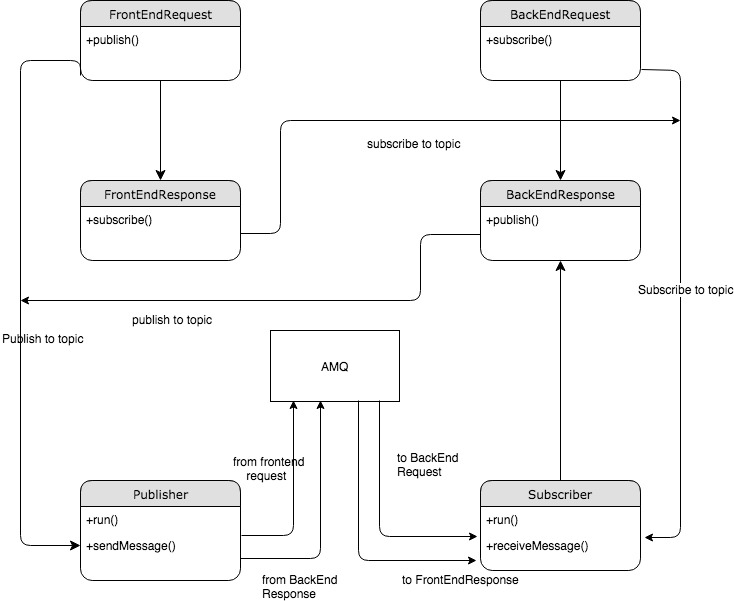
The overall design diagram for the ActiveMQ consists of the following components:

* FrontEnd Request
* FrontEnd Response
* Publisher
* AMQ
* Subscriber
* BackEnd Request
* BackEnd Response



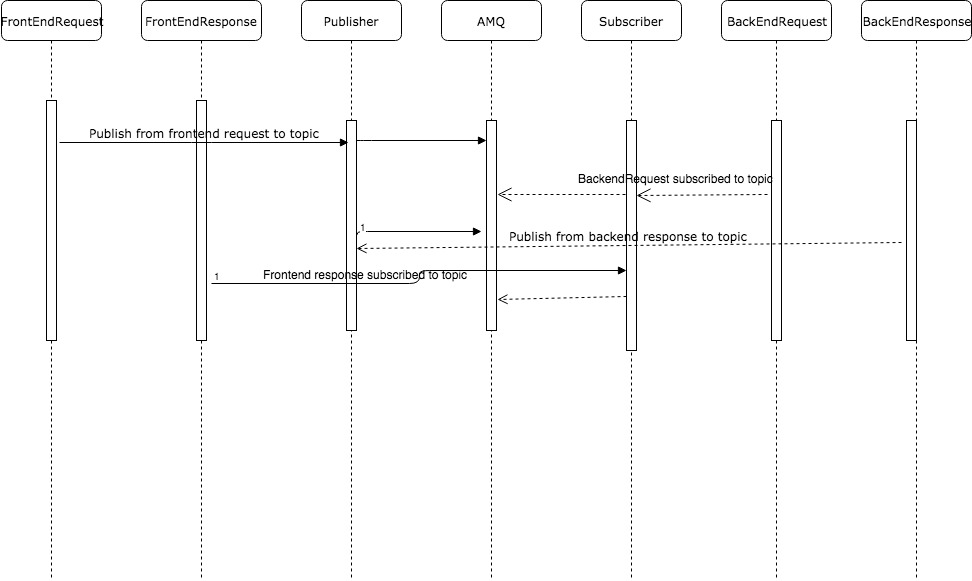
1. **Class Diagram:**

The followingclass diagram describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. The arrow represents the association link between two classes.



1. **Sequence Diagram:**

The sequence diagram below describes the connections between various components involved in the design:



1. **Sequence Diagram 2:**

The following sequence diagram describes a part of the architecture on how the ASM image is handled, when the assembler and the simulator is used. It describes how the correct ASM image is verified with the id associated with it, thereby reducing the latency of sending the ASMImage back and forth. In this diagram, FEEventAgent is (FrontEndRequest + FrontEndResponse) and BEEventAgent is (BackEndRequest + BackEndResponse).

