Asynchronous Communication and Real-Time Notification

Assignment 2

Distributed Systems

**Student:** Jimborean Oana-Elena

**Group:** 30443

1. **Introduction:**

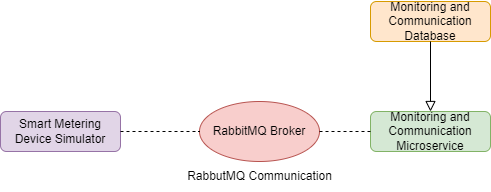
I had to develop a Monitoring and Communication Microservice for the Energy Management System.The microservice is based on a message broker middleware that gathers data from the smartmetering devices, processes the data to compute the hourly energy consumption and stores it inthe database of the Monitoring and Communication Microservice.

The synchronization between the databases of Device Management Microservice and the new Monitoring and Communication Microservice is made through an event-based system that uses a topic for device changes (sends device information through a queue for the Monitoring and Communication Microservice).

A Smart Metering Device Simulator application is implemented as the Message Producer. It simulates a smart meter by reading energy data from a sensor.csv file and sends data in the form < timestamp, device\_id, measurement\_value > to the Message Broker (i.e., a queue). The timestamp is taken from the local clock, and the device\_id is unique to each instance of the Smart Metering Device Simulator and corresponds to the device\_id of a user from the database (as defined in Assignment 1).

1. **Architecture**

The system architecture consists of a Consumer (Java Spring microservice) for monitoring the energy consumption of devices, having its dedicated MySQL database and a Producer (Smart Metering Device Simulator), also implemented in Java Spring, which simulates the data reading and data sending as messages to the Consumer. When the applications are started, the Simulator generates three types of data: a timestamp from the actual time-zone, a deviceId and a measurement value which is read from a .csv file. This data is send to a queue through RabbitMQ and received by the Consumer. The Consumer takes the data, it displays it in the console and it also saves it to the database.



1. **Backend Microservice (Consumer)**

Monitoring and Communication Microservice**:** a message broker middleware to gather real-time data from smart metering devices. This microservice handles the data, computes the hourly energy consumption, and takes charge of storing this computed information in the dedicated database and also displaying the data in the console. The consumer component within the Monitoring and Communication Microservice is responsible for persisting hourly energy values into the Monitoring database.

1. **Smart Metering Device Simulator (Producer)**

Smart Metering Device Simulator: serves as a standalone desktop application designed to emulate a smart meter's functionality. Its primary purpose is to simulate the behavior of a smart meter by generating energy data sourced from a sensor.csv file. This application mimics the behavior of a real smart meter, generating and dispatching energy data packets to the Message Broker in a simulated real-time manner.

1. **Database**

**Monitoring Database**: stores the data from the smart metering devices. The total hourly energy consumption is computed and saved into the database.

1. **Deployment**

To deploy the Monitoring and Communication System, I used Docker, which provides the network and container virtualization functionality, allowing the containers to interact with each other in a secure environment and allowing certain ports and endpoints to pass through to the clients.

Backend deployment: the microservice was containerized, alongside its database and RabbitMQ, using Docker images which encapsulate the applications for consistent deployment.

