MINISTRY OF EDUCATION

TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA, ROMANIA
FACULTY OF AUTOMATION AND COMPUTER SCIENCE
COMPUTER SCIENCE DEPARTMENT

DISTRIBUTED SYSTEMS

Assignment 2

Asynchronous Communication

Prof. Tudor Cioara
As. Liana Toderean
As. Antonesi Gabriel

S.l. Marcel Antal
As. Alexandru Rancea

Conf. Cristina Pop
As. Dan Mitrea

2025-2026

1. Requirements

Design and implement a **Monitoring Microservice** for the Energy Management System. This microservice will process data from smart metering devices, compute the **hourly energy consumption**, and store the results in its dedicated database. The architecture extends the system from Assignment 1, introducing asynchronous communication and device data simulation.
The solution will contain the following new components:

- **Device Data Simulator (Producer):** A standalone desktop application that simulates smart meter readings by generating energy consumption values at 10-minute intervals. Each generated measurement has the form `<timestamp, device_id, measurement_value>` and is sent to the message broker in JSON format.
- **Message-Oriented Middleware (RabbitMQ):** A message broker used to decouple data producers (simulators) from consumers (the Monitoring Microservice).
- **Monitoring Microservice (Consumer):** Subscribes to the queue, processes incoming device data, computes hourly energy totals, stores the results in the Monitoring Database.

To ensure consistency across all microservices, the solution must extend Assignment 1 with an **event-based synchronization mechanism** using RabbitMQ.

- All synchronization events (both **users** and **devices**) must be sent to a **dedicated Synchronization Queue**.
- When a new **user** is added through the User Management Microservice, the service must publish a synchronization message to this queue containing the user's ID and (if any) other relevant attributes. Every microservice that stores user-related data (Device Microservice) must consume this message and insert the user's ID in its local database.
- When a new **device** is added through the Device Management Microservice, the service must publish a synchronization message to the same queue containing the device's ID and relevant attributes. All microservices that manage device-related data must consume this message and insert the device's ID (or other relevant information) into their associated database.

1.1. Component description:

1. Device Data Simulator

- Standalone application that generates synthetic measurements every 10 minutes and sends them as JSON data to a RabbitMQ queue;
- The data is generated using the following criteria:
    - Each simulated smart meter reading represents the energy consumed over a 10-minute interval. The simulator starts from a random base load and adjusts it using simple calculations, mimicking realistic patterns (as much as possible); example: lower consumption during the night, higher in the evening, small fluctuations in-between. These generated values are then sent as JSON messages to the message broker, using the structure defined above.

2. Message Brokers – Synchronization Broker, Data Collection Broker

- Ensures decoupling between producers and consumers;
- Handles queues for data and synchronization events.

3. Monitoring Microservice

- Acts as a message consumer, subscribing to the queue and processing incoming sensor data;
- Aggregates device values into hourly totals and inserts them in the Monitoring database.

1.2. Implementation technologies:

- We recommend RabbitMQ for message brokers, REST for microservices (Java Spring REST or .NET Web API), for storing data we suggest using PostgreSQL or MySQL and for reverse proxy, we suggest Traefik, especially if you already defined it during the first assignment. Docker is (still) **mandatory** for the deployment of the new microservice.

2. Deliverables

- A solution description document containing:
  a) UML Deployment diagram.
  b) Readme file containing build and execution considerations.
- Source files. The source files and the database dump will be uploaded onto the personal *gitlab* account created at the *Lab resources* laboratory work, following the steps:
  - Create a repository on *gitlab* with the exact name: *DS2025_Group_LastName_FirstName_Assignment_Number*
  - Push the source code and the deployment diagram (push the code not an archive with the code or war files)
  - Share the repository with the user *utcn_dsrl*

3. Evaluation

3.1. Assignment Related Basic Questions:

During assignment evaluation and grading you will be asked for details about the following topics:

- Message Oriented Middleware types; Queue vs Topic; Point-to-Point vs Publish-Subscribe communication.
- Additional concepts discussed during the previous laboratory sessions

3.2. Grading

- The **assignment** will be graded as follows:

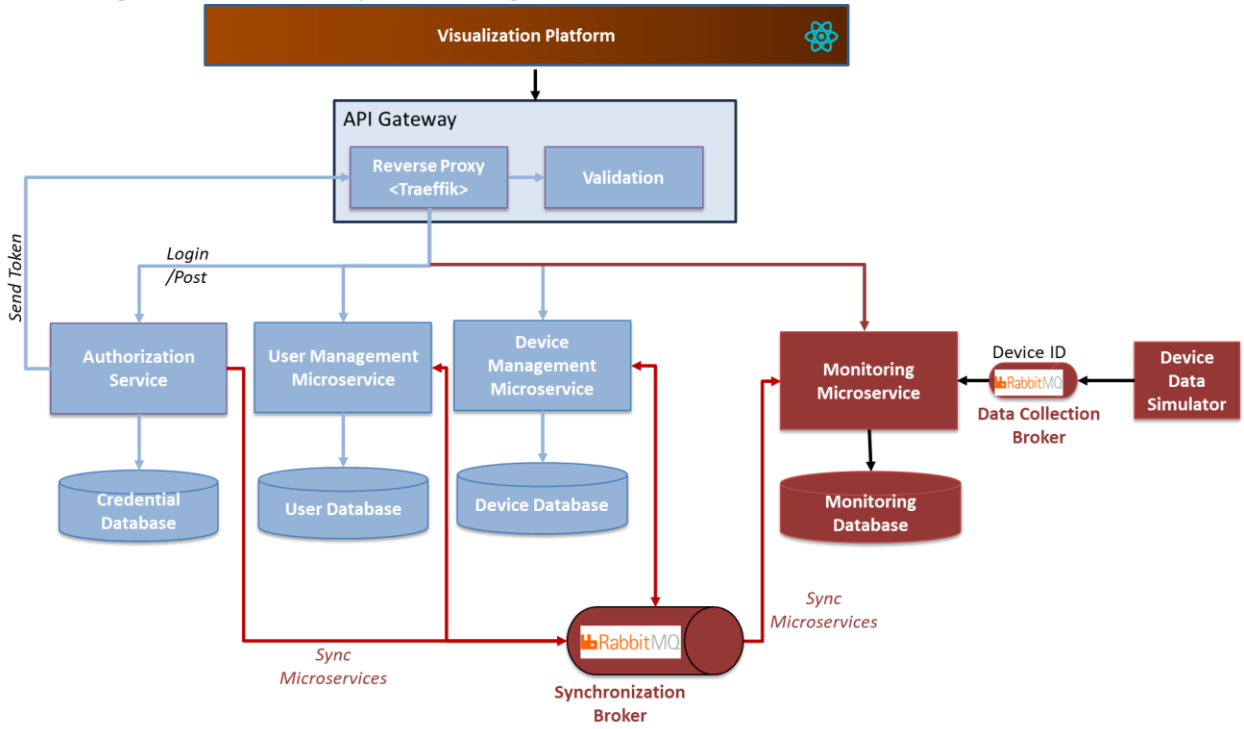
Points	Requirements
5 p	Minimum to pass: • Monitoring Microservice with message consumer component • Implement the Message producer and Message broker • Insert hourly energy consumption data in the database • Readme file • Correct answers to 3.1 questions
2 p	User Synchronization
1 p	Device Synchronization
2 p	A client can view: his/her historical energy consumption as line charts or bar charts for one day (OX- hours; OY- energy value [kWh] for that hour). Select day from a calendar

- The **project** will be graded as follows:

Points	Requirements
10 p	Minimum to pass: • Reverse Proxy • Docker deployment • Deployment diagram

***NOTE:** The sensor application should have a configuration file where you can set the device ID for the client for which you want test the application for.

4. Assignment conceptual diagram



5. Bibliography

1. https://dsrl.eu/courses/sd/
2. Lab Book: I. Salomie, T. Cioara, I. Anghel, T.Salomie, *Distributed Computing and Systems: A practical approach*, Albastra, Publish House, 2008, ISBN 978-973-650-234-7
3. Lab Book: M. Antal, C. Pop, D. Moldovan, T. Petrican, C. Stan, I. Salomie, T. Cioara, I. Anghel, Distributed Systems – Laboratory Guide, Editura UTPRESS Cluj-Napoca, 2018 ISBN 978-606-737-329-5, 2018, https://biblioteca.utcluj.ro/files/carti-online-cu-coperta/329-5.pdf
4. https://www.rabbitmq.com/documentation.html