

Sisteme distribuite

Ciochină Cătălina-Andreea

Grupa 30643

Anul 2024-2025

Assignment 2

Asynchronous Communication and Real-Time Notification

Prof. coord. Colceriu Tudor

Cuprins:

- 1. Introducere***
- 2. Arhitectura Conceptuală***
- 3. Diagrama de deployment UML***
- 4. Detalii de Implementare***
- 5. Considerații de Securitate***
- 6. Fișier Readme***
- 7. Concluzie***

1. Introducere

Sistemul de Management al Energiei (EMS) este o aplicație web destinată gestionării conturilor utilizatorilor și a dispozitivelor inteligente de măsurare a energiei. Proiectul utilizează Spring Boot pentru construirea microserviciilor: `user_ms`, `device_ms` și `monitoring_ms`, și Angular pentru dezvoltarea frontend-ului. Sistemul permite utilizatorilor cu rol de administrator să efectueze operațiuni CRUD asupra utilizatorilor și dispozitivelor, în timp ce utilizatorii clienți pot vizualiza dispozitivele asociate contului lor și monitoriza consumul de energie. Un simulator este inclus în arhitectură pentru a genera și trimite date de consum energetic către microserviciul de monitorizare.

2. Arhitectura Conceptuală

Arhitectura conceptuală a sistemului cuprinde următoarele componente:

Frontend (Angular)

- O aplicație web dezvoltată cu Angular care permite interacțiunea utilizatorilor cu sistemul.

Microservicii (Spring Boot)

1. Microserviciul de gestionare a utilizatorilor (`user_ms`):

→ Gestionează operațiunile CRUD pentru utilizatori și autentificarea acestora.

2. Microserviciul de gestionare a dispozitivelor (`device_ms`):

→ Gestionează dispozitivele de energie și asocierile lor cu utilizatorii.

3. Microserviciul de monitorizare a consumului energetic (`monitoring_ms`):

→ Colectează și procesează datele de consum energetic generate de simulator.

Simulator de Date Energetice

- Un modul dedicat care simulează generarea de date de consum energetic pentru dispozitive. Aceste date sunt trimise periodic către **`monitoring_ms`**.

Baza de date (PostgreSQL)

- Fiecare microserviciu își menține propria bază de date:
 - Baza de date pentru utilizatori.
 - Baza de date pentru dispozitive.
 - Baza de date pentru monitorizare.

Prezentare generală la nivel înalt:

- **Autentificarea utilizatorilor:**

Utilizatorii se conectează la sistem, iar credențialele lor sunt validate. În funcție de rol (administrator sau client), aceștia au acces la diferite funcționalități.

- **Controlul accesului bazat pe roluri:**

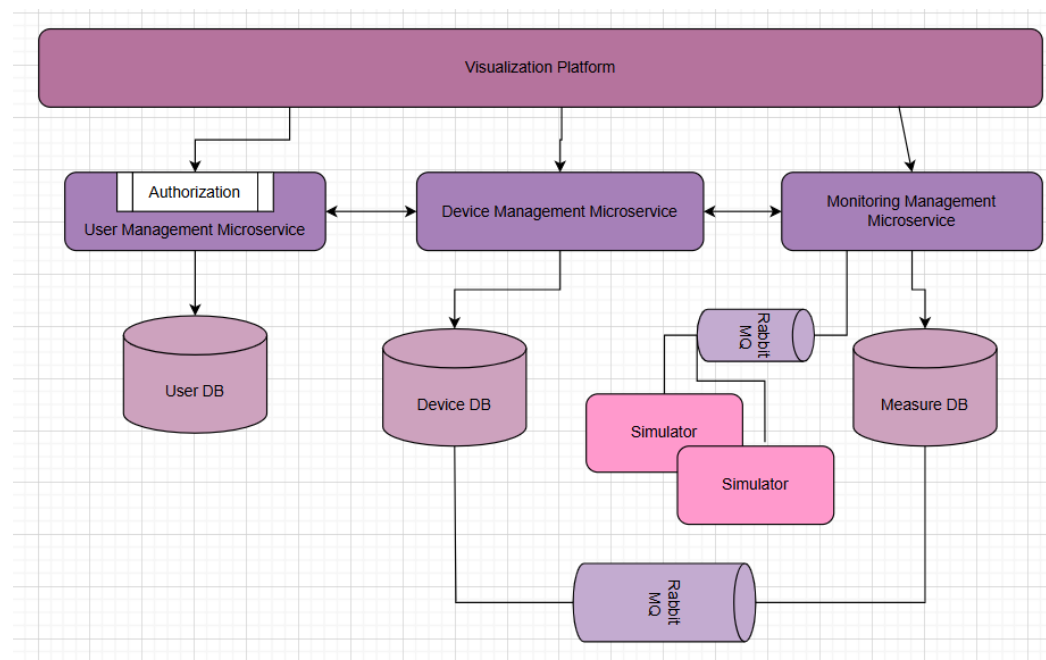
Funcționalitățile sunt restricționate în funcție de rolurile utilizatorilor.

- **Monitorizarea consumului:**

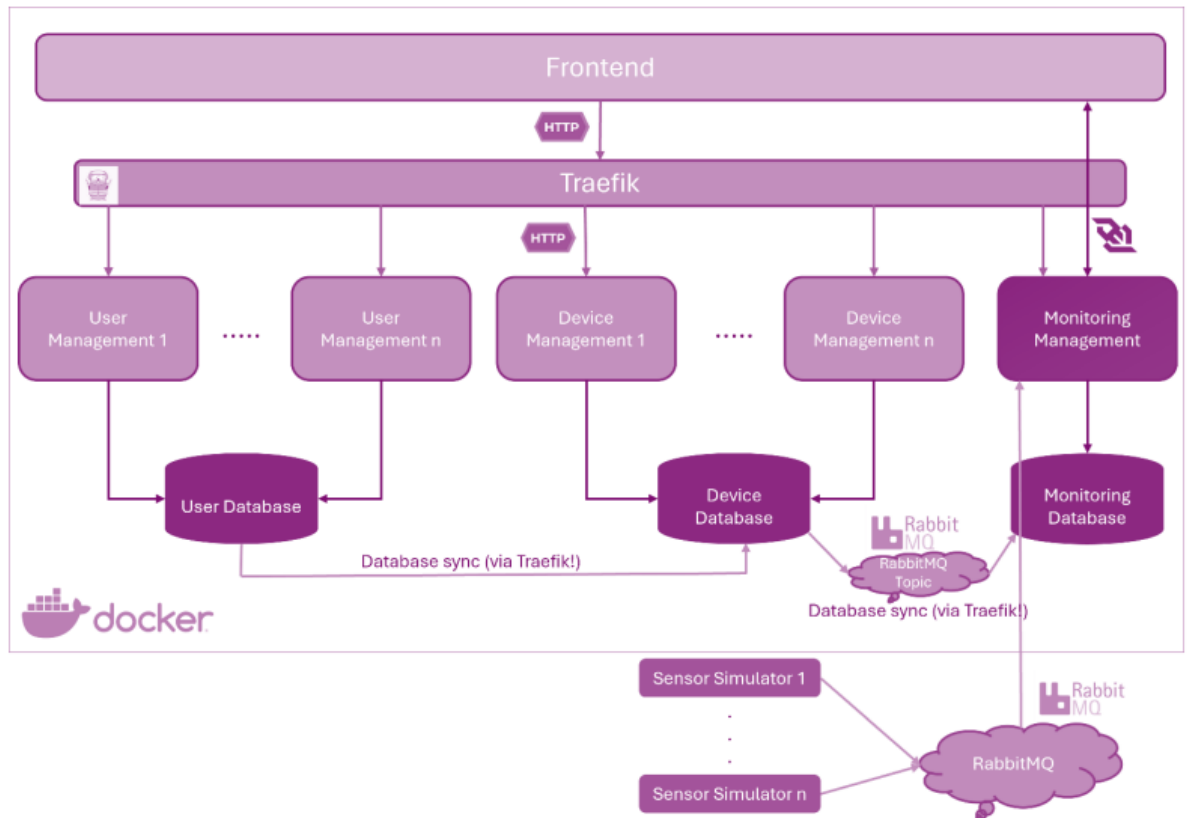
Utilizatorii pot vizualiza datele de consum energetic generate și procesate în timp real de simulator și microserviciul de monitorizare.

- **Comunicația microserviciilor:**

Microserviciile comunică prin API-uri REST pentru operațiuni precum asocierea utilizator-dispozitiv și înregistrarea datelor de consum.



3. Diagrama de Deployment UML



4. Detalii de Implementare

Tehnologii Utilizate

- **Frontend:** Angular
- **Microservicii:** Spring Boot
- **Baza de date:** PostgreSQL
- **Securitate:** Spring Security

Docker

Aplicația este containerizată utilizând **Docker** și orchestrată cu Traefik pentru rutare. Arhitectura include un frontend Angular (port 4200), trei microservicii backend pentru gestionarea utilizatorilor, dispozitivelor și măsurătorilor (porturi 8084, 8081, 8082), și trei baze de date PostgreSQL dedicate fiecărui microserviciu. Toate serviciile sunt conectate printr-o rețea Docker personalizată (custom_network) pentru comunicație internă securizată.

Traefik funcționează ca proxy reversibil, gestionând rutele prin hostname-uri (user.localhost, device.localhost, etc.) și asigurând suport pentru WebSocket și CORS. Microserviciile comunică cu bazele de date utilizând variabile de mediu pentru configurare, iar frontend-ul trimite cereri către backend prin hostname-urile configurate în Traefik. Această abordare facilitează scalabilitatea și izolarea componentelor.

5. Considerații de Securitate

Pentru a asigura securitatea sistemului, au fost implementate următoarele măsuri folosind **Spring Security**:

1. Autentificare:

- Implementarea autentificării prin email și password.
- Utilizarea token-urilor JWT (JSON Web Tokens) pentru gestionarea sesiunilor de utilizator.
- Configurarea endpoint-urilor pentru autentificare și protejarea celorlalte endpoint-uri în funcție de roluri.

2. Controlul accesului bazat pe roluri:

- Restricționarea accesului la funcționalități în funcție de rolurile utilizatorilor (admin/client).
- Definirea regulilor de acces în configurația Spring Security pentru a asigura că utilizatorii nu pot accesa resursele care nu le sunt permise.

6. Fișier Readme

Atașat în repository.

7. Concluzie

Sistemul de Management al Energiei (EMS) reprezintă o soluție modernă și scalabilă pentru gestionarea utilizatorilor, dispozitivelor inteligente și monitorizarea consumului de energie. Utilizând o arhitectură bazată pe microservicii și tehnologii precum Spring Boot, Angular și PostgreSQL, sistemul oferă o separare clară a responsabilităților și o modularitate crescută.

Adăugarea simulatorului și a microserviciului de monitorizare asigură colectarea automată și procesarea eficientă a datelor energetice, oferind utilizatorilor informații în timp real. Implementarea cu containere Docker asigură portabilitatea și uniformitatea în toate mediile de dezvoltare și producție.