

# Integrated Energy Management System

Popovici Eusebiu-Ionut

## 1 Introducere

Acest proiect se axează pe dezvoltarea unei aplicații dedicate monitorizării și gestionării consumului de energie. Soluția oferă o interfață intuitivă pentru vizualizarea utilizatorilor și a dispozitivelor asociate, precum și pentru colectarea și procesarea datelor de consum energetic. Aplicația folosește React pentru frontend, iar backend-ul este construit utilizând tehnologii precum Spring Boot, RabbitMQ și baze de date MySQL.

## 2 Structura aplicației

- **\*\*Backend-ul\*\*** constă într-un microserviciu responsabil cu procesarea datelor de la dispozitive. Acesta primește măsurători prin RabbitMQ, calculează consumul orar de energie, pregătește informațiile pentru afișare grafică și trimite notificări în timp real utilizatorilor prin WebSocket atunci când se depășesc limitele de consum stabilite. Toate datele sunt stocate într-o bază de date MySQL.
- **\*\*Simulatorul\*\*** este o aplicație separată care încarcă periodic valori dintr-un fișier CSV ('sensor.csv') și transmite datele sub formă de mesaje JSON către RabbitMQ. Fiecare mesaj conține un timestamp, un device\_id unic și valoarea măsurată.
- **\*\*Integrarea containerizată\*\*** se realizează cu ajutorul Docker. Fiecare componentă utilizează propriul Dockerfile, iar orchestrarea se face printr-un fișier 'docker-compose'. Aplicația poate

fi rulată din consolă, iar testarea este realizată din interfața React.

### 3 Implementare

1. **\*\*Simulatorul\*\*** este implementat ca o aplicație Spring Boot care citește valorile din fișierul CSV ('sensor.csv'). Utilizatorul poate iniția simularea printr-un buton din interfață. Mesajele sunt trimise către RabbitMQ la un interval de 1 secundă folosind adnotarea '@Scheduled'. Fiecare mesaj include timestamp-ul, device\_id-ul și valoarea consumului măsurat.
2. **\*\*Microserviciul de Monitorizare\*\*** preia mesajele din RabbitMQ și procesează fiecare măsurătoare. În cazul în care valoarea măsurată, împreună cu media ultimelor 5 valori, depășește consumul permis, se trimite o notificare utilizatorului prin interfața React. De asemenea, microserviciul sincronizează modificările bazelor de date utilizând RabbitMQ și un 'topicExchange'. Modificările sunt transmise folosind un 'routing key' specific tipului de operațiune (adăugare, actualizare, ștergere). Utilizatorii pot selecta o zi din calendar pentru a vizualiza grafic consumul aferent acelei perioade.
3. **\*\*Notificări în timp real\*\***: În cazul în care consumul de energie depășește limitele stabilite, aplicația trimite notificări utilizatorilor în timp real. Aceste notificări sunt generate prin WebSocket și afișate direct în interfața React. Acest mecanism ajută utilizatorii să ia măsuri rapide pentru a reduce consumul.
4. **\*\*Vizualizarea datelor de consum\*\***: Consumul de energie pentru o anumită zi poate fi analizat folosind o serie de grafice generate automat. Utilizatorii selectează ziua dorită dintr-un calendar și aplicația afișează consumurile orare în detaliu. Graficele oferă o reprezentare vizuală clară a fluctuațiilor consumului de energie, ajutând la identificarea tiparelor.

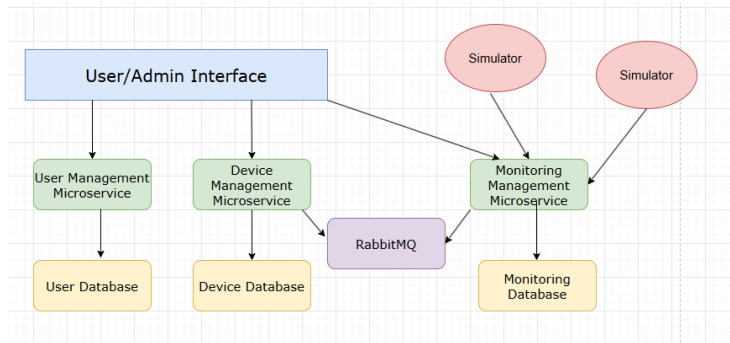


Figure 1: Diagrama conceptuală a arhitecturii

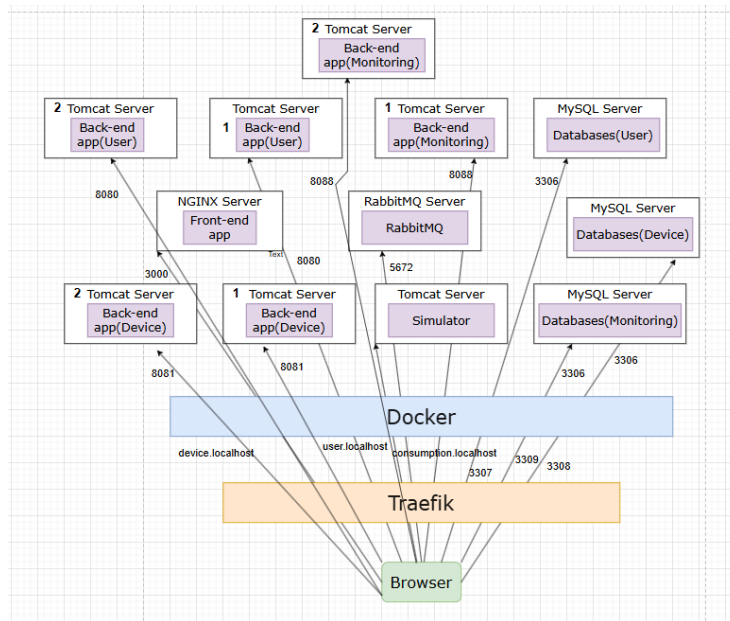


Figure 2: Diagrama de deployment utilizând Docker

## 4 Concluzie

Soluția propusă pentru monitorizarea și gestionarea consumului energetic aduce beneficii semnificative prin integrarea eficientă a tehnologiilor moderne. Utilizatorii dispun de o interfață simplă și intuitivă

pentru a analiza consumul energetic și a primi notificări în timp real atunci când depășesc limitele stabilite.

Microserviciul de Monitorizare procesează rapid și precis datele primite de la simulator, sincronizează automat modificările între baze de date și folosește eficient resursele prin RabbitMQ. Simulatorul oferă flexibilitate în generarea scenariilor de consum energetic, facilitând testarea și adaptarea aplicației la diverse nevoi.

Containerizarea cu Docker asigură o implementare ușoară, scalabilă și sigură. Comunicarea între microservicii este gestionată prin Traefik și un network dedicat, garantând performanță și fiabilitate. Utilizarea RabbitMQ și WebSocket sporește interactivitatea și eficiența, făcând din această aplicație o soluție completă pentru managementul integrat al energiei.