Energy Management System

Assignment 3

DISTRIBUTED SYSTEMS

Zubascu Ileana

TI an 4 grupa 4

Introducere

Partea a treia a proiectului se concentrează pe implementarea mecanismului de securitate și autorizare folosind **Bearer Token** și **JWT (JSON Web Token)**, care protejează toate endpoint-urile microserviciilor. Prin această abordare, accesul la resursele sensibile ale aplicației este restricționat pe baza rolurilor utilizatorilor, asigurându-se astfel că doar utilizatorii autorizați pot accesa anumite funcționalități.

Autentificarea se face printr-un token care este transmis la fiecare cerere, iar acest token conține informații despre utilizator, inclusiv rolul acestuia (de exemplu, admin sau user). Sistemul utilizează un mecanism de validare a token-urilor la fiecare cerere, iar în funcție de rolul atribuit, utilizatorii sunt direcționați către resursele corespunzătoare. De asemenea, implementarea **JWT** asigură că token-urile sunt sigure și dificil de falsificat, iar sesiunile de utilizator sunt gestionate eficient, fără a necesita păstrarea unor sesiuni pe server.

Partea a doua a proiectului se axează pe implementarea unui **microserviciu de chat** care permite comunicarea în timp real între utilizatori și admini. În acest microserviciu, utilizatorii pot iniția conversații cu adminul, iar adminul poate comunica simultan cu mai mulți utilizatori. Comunicarea se realizează prin intermediul **WebSocket**, care asigură un canal de comunicare bidirecțional, rapid și eficient.

Un aspect esențial al acestui microserviciu este **funcționalitatea de typing**, care adaugă o dimensiune interactivă și dinamică experienței de chat. Astfel, utilizatorii pot vedea în timp real atunci când persoana cu care comunică tastează un mesaj, ceea ce îmbunătățește semnificativ interacțiunea și face conversațiile mai fluide și mai naturale.

Microserviciul nu stochează mesajele pe termen lung, iar acestea sunt disponibile doar pe durata deschiderii conversației, urmând să fie șterse odată ce sesiunea de chat este închisă. Această abordare asigură un sistem simplu și eficient, fără necesitatea gestionării unui sistem de stocare a mesajelor pe termen lung, reducând astfel complexitatea și costurile de întreținere ale aplicației.

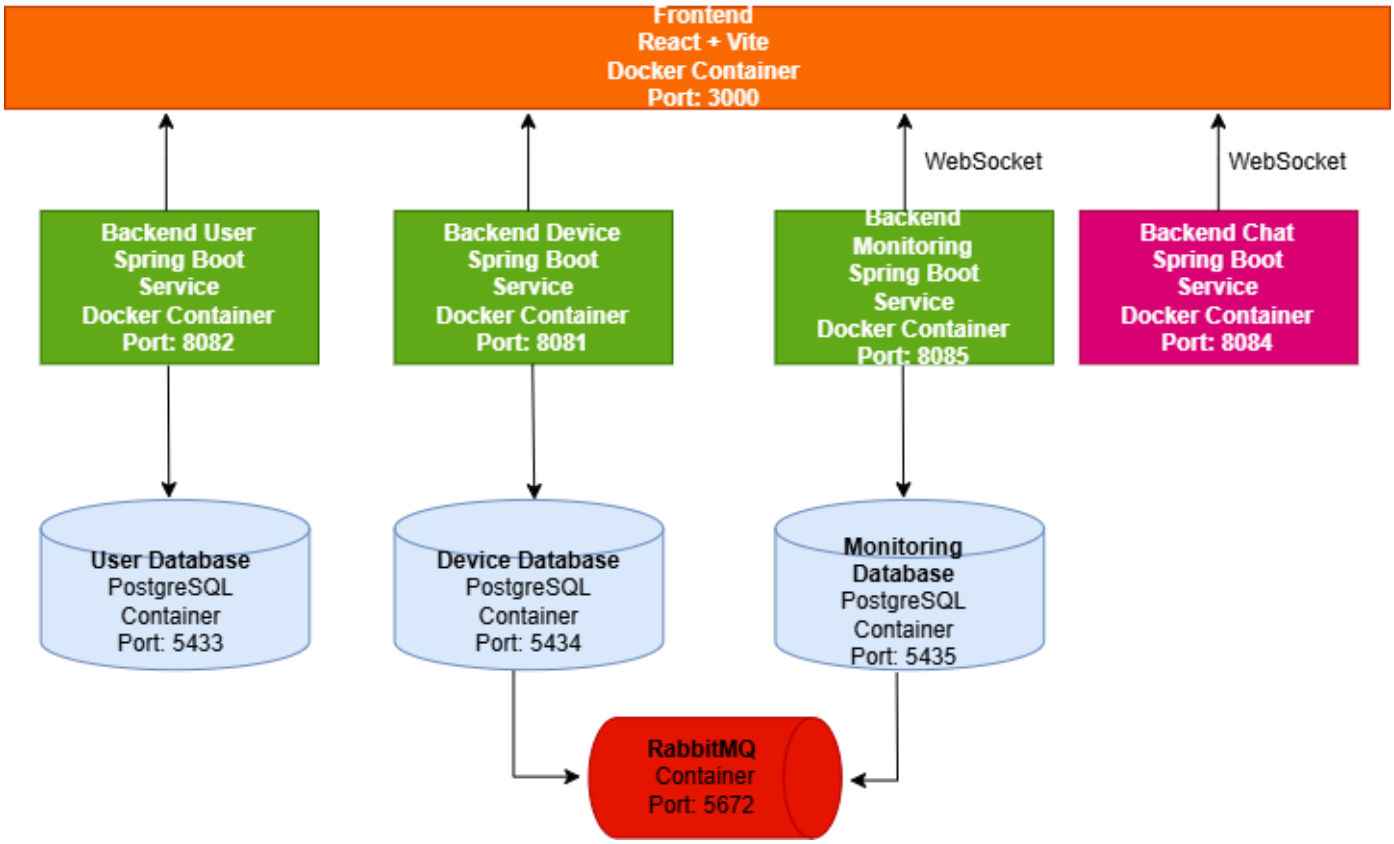
Deploy-ul este realizat în **Docker** folosind 9 containere: baze de date pentru utilizatori, device-uri și monitoring, backend-uri pentru fiecare microserviciu, frontend, RabbitMQ pentru mesagerie, și **Traefik** pentru load balancing și reverse proxy. RabbitMQ este integrat în microserviciul de monitoring, care gestionează coada de mesaje pentru sincronizare și notificări. Arhitectura asigură scalabilitate și comunicare eficientă între componente.. Fiecare container rulează pe un port separat, permițând distribuirea și comunicarea clară între componentele aplicației.

Flow-ul Aplicației

Aplicația implementează un flux distinct pentru administrator și pentru utilizatorul obișnuit, asigurând acces diferențiat în funcție de permisiunile asociate fiecărui cont.

1. **Fluxul pentru Admin**:
    - Administratorul trebuie să se autentifice folosind datele sale de acces cu permisiuni de administrare. După introducerea corectă a credentialelor, acesta este redirecționat către pagina /admin.
    - Pe această pagină, sunt afișate două butoane: **Devices** și **Users**. Fiecare buton oferă acces la secțiuni diferite ale aplicației:
        - **Devices**: Selectând acest buton, administratorul poate vizualiza toate dispozitivele existente, împreună cu detalii precum descrierea, adresa, energia maximă și userId-ul asociat. Administratorul are, de asemenea, posibilitatea de a efectua operațiuni de CRUD (creare, citire, actualizare, ștergere) pe aceste dispozitive.
        - **Users**: Prin acest buton, administratorul poate gestiona utilizatorii aplicației, efectuând operațiuni similare de CRUD. De asemena, exista un buton de **Chat** in care adminul poate initia o conversatia cu un user specific
2. **Fluxul pentru Utilizator**:
    - Utilizatorul obișnuit, autentificat cu un cont de utilizator, este redirecționat către pagina /user. Aici, acesta poate vizualiza doar dispozitivele la care a fost asociat, fără a avea permisiuni de modificare sau acces la detalii administrative.
    - Utilizatorul poate iniția o simulare pentru un dispozitiv asociat, care monitorizează consumul orar de energie. Dacă se depășește limita maximă, utilizatorul este notificat în timp real. Simularea poate fi oprită manual de către utilizator.
    - Userul are la dispozitie un chat in care poate comunica cu adminul
3. **Controlul Accesului**:
    - Paginile /admin și /user sunt protejate prin mecanisme de autorizare, astfel că un utilizator nu poate accesa aceste pagini decât dacă are autoritatea necesară, bazată pe rolul și permisiunile definite în token-ul JWT asociat fiecărui cont.

UML diagram



Deployment Diagram