

Sisteme distribuite

Energy Management System

Assignment 3

Nume: Muntean Diana-Gratiana

Grupa: 30641

2024-2025

Cuprins:

1) Conceptual architecture of the distributed system

2) UML Deployment diagram

3) Readme file containing build and execution considerations

1. Conceptual architecture of the distributed system

Platforma de management al energiei permite accesul prin autentificare pentru două tipuri de utilizatori: administrator și client. Administratorul are acces la operații CRUD pentru conturi de utilizator și dispozitive inteligente. Fiecare utilizator își poate vizualiza dispozitivele asociate și consumațiile de energie.

Sistemul este o aplicație web pentru managementul energiei, construită pe o arhitectură bazată pe microservicii:

* **Frontend**: Implementat în React pentru a permite o interfață dinamică, accesibilă atât pentru administratori, cât și pentru clienți.
* **Microservicii**:
  + **User Management Microservice**: Gestionarea utilizatorilor prin operații CRUD.
  + **Device Management Microservice**: Administrarea dispozitivelor de măsurare inteligentă prin operații CRUD.
  + **Chat Management Microservice**: Aplicația de chat permite comunicarea asincronă între utilizatori și administrator, susținând mesaje bidirecționale, conversații simultane cu mai mulți utilizatori și notificări pentru citirea și scrierea mesajelor.
* **Securitate**: Autentificarea bazată pe sesiuni/cookies pentru acces controlat, oferind fiecărui rol acces limitat.

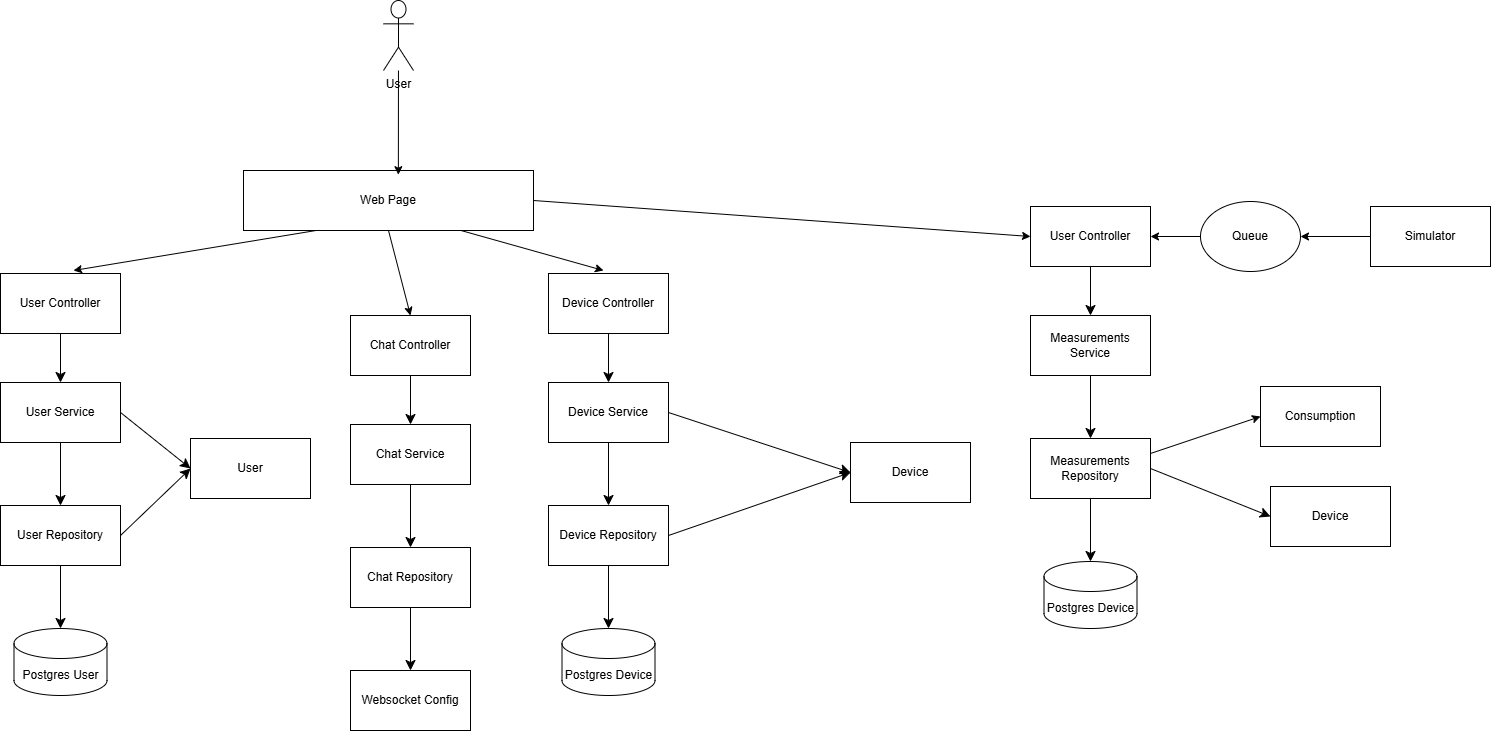
**Autentificare JWT pentru Securitatea API-ului**

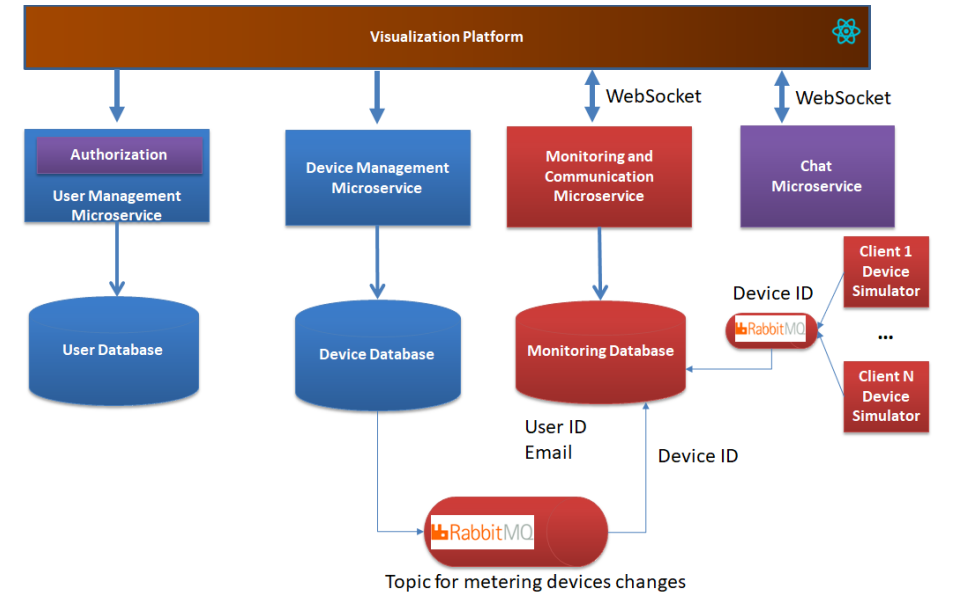
**JSON Web Token (JWT)** este un mijloc compact și sigur pentru reprezentarea declarațiilor ce trebuie transferate între două părți. Este adesea utilizat pentru autentificare și autorizare în dezvoltarea web.  
În această aplicație Spring Boot, JWT este folosit pentru a autentifica utilizatorii care accesează API-ul REST expus. Doar utilizatorii cu un JWT valid pot utiliza API-ul. Acest lucru adaugă un strat suplimentar de securitate aplicației.

**Pașii pentru autentificarea JWT:**

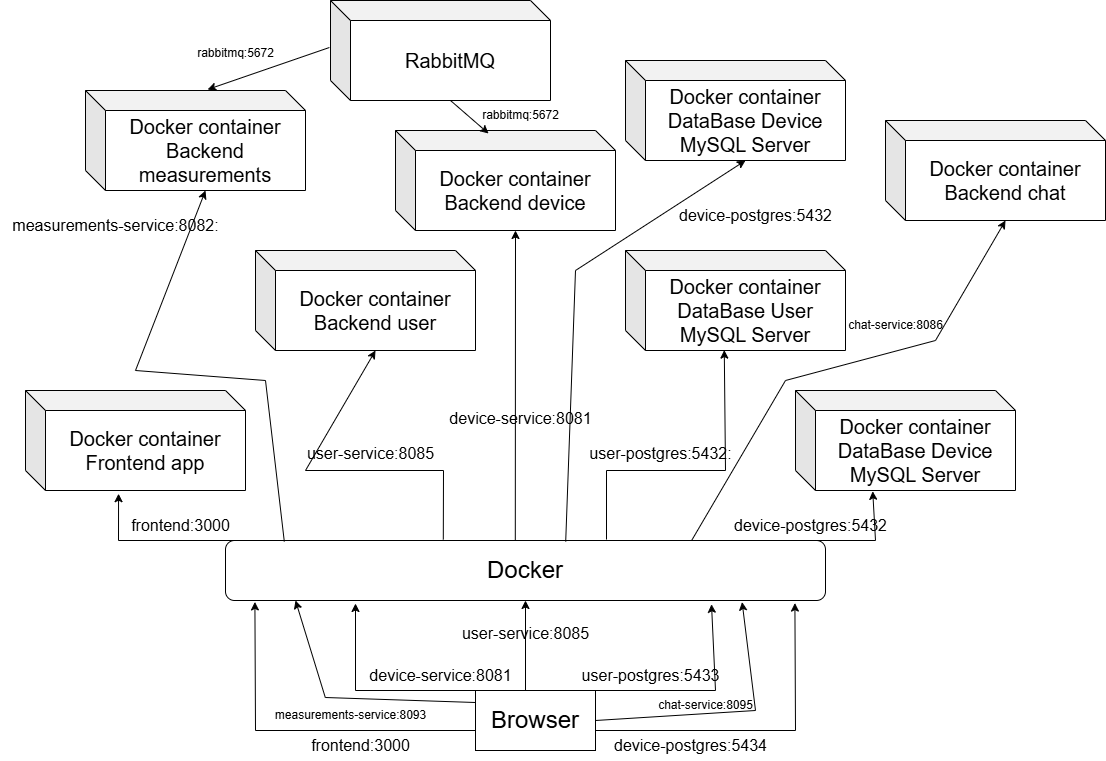
* **Autentificarea utilizatorului:** Utilizatorii sunt autentificați prin furnizarea unor credențiale valide.
* **Generarea JWT:** La autentificarea cu succes, un JWT este generat pentru utilizator.
* **Acces API:** Utilizatorii pot accesa API-ul incluzând JWT-ul generat în cererile lor.

**Validarea JWT:** API-ul validează JWT-urile primite pentru a asigura autenticitatea utilizatorului





1. UML Deployment diagram



1. Readme file containing build and execution considerations

Instrucțiuni privind configurarea și rularea aplicației:

* **Dependențe**: Java 17, Node.js 16, PostgreSQL pentru baza de date.
* **Pași de rulare**:
  + Clonarea repository-urilor din GitLab.
  + Rularea fiecărui microserviciu în IntelliJ și a frontend-ului cu npm start în terminal.
  + Rularea cu docker: docker-compose up --build