

Sisteme distribuite

Ciochină Cătălina-Andreea

Grupa 30643

Anul 2024-2025

Assignment 1

Request-Reply Communication

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 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.

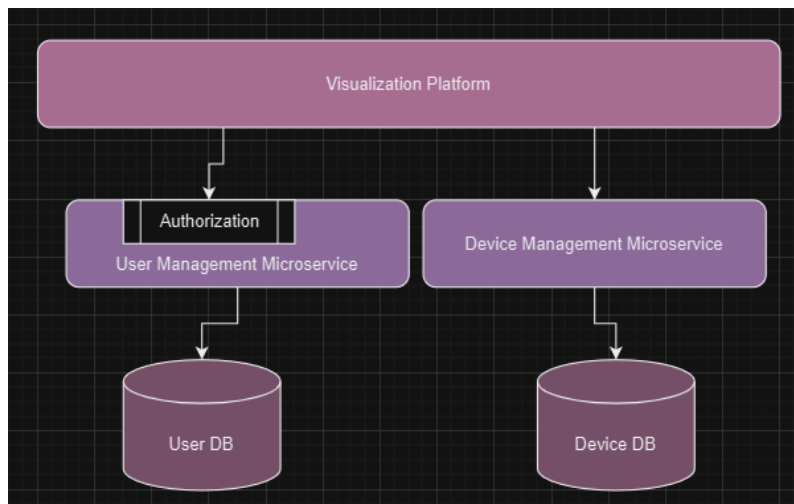
2. Arhitectura Conceptuală

Arhitectura conceptuală a sistemului cuprinde următoarele componente:

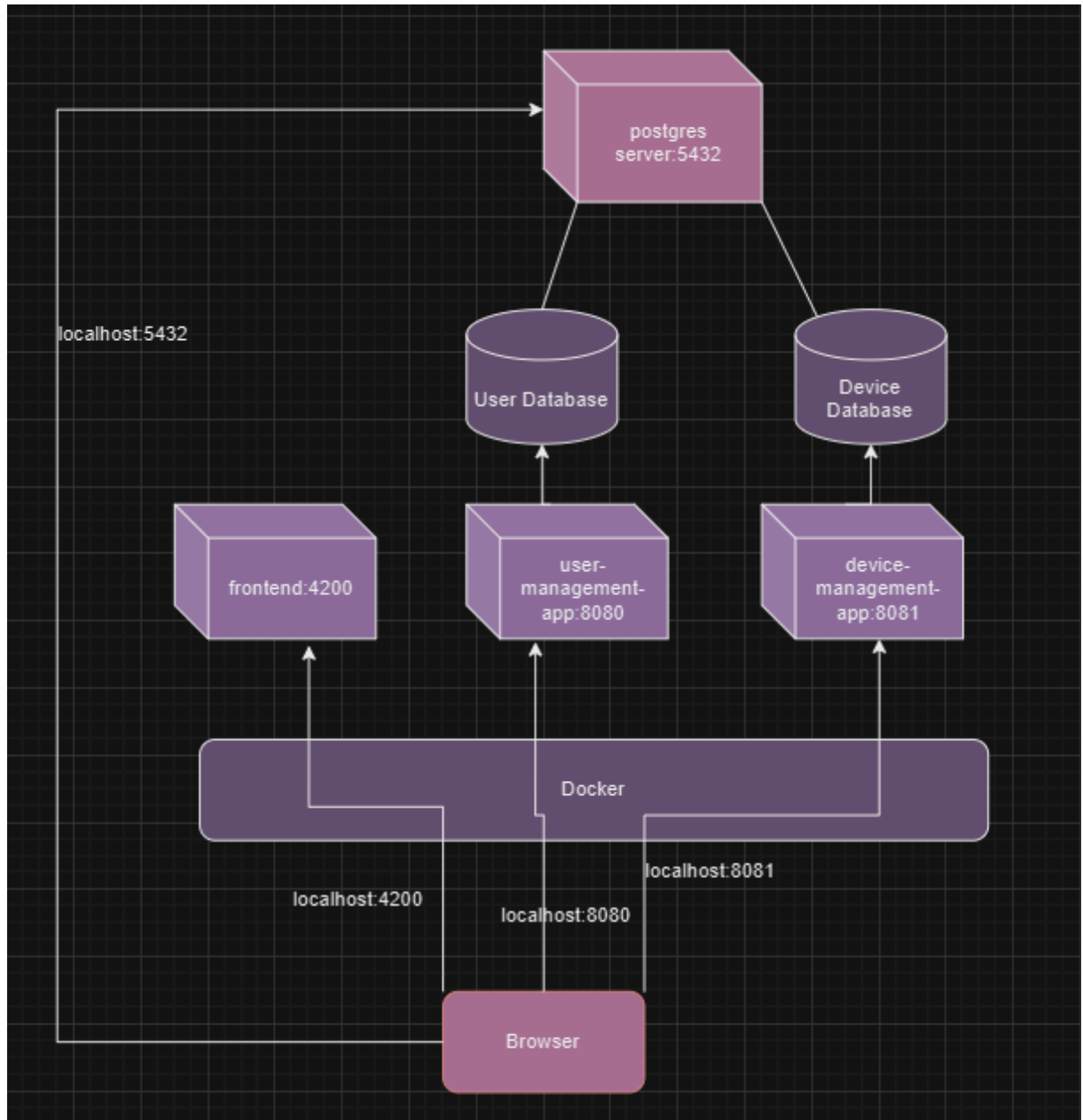
1. **Frontend (Angular):** O aplicație web dezvoltată cu Angular care permite interacțiunea utilizatorilor cu sistemul.
2. **Microservicii:**
 - **Microserviciul de gestionare a utilizatorilor:** Implementat cu Spring Boot, gestionează operațiunile CRUD pentru utilizatori și autentificarea acestora.
 - **Microserviciul de gestionare a dispozitivelor:** De asemenea, implementat cu Spring Boot, gestionează dispozitivele de energie și asocierile cu utilizatorii.
3. **Baza de date:** Fiecare microserviciu își menține propria bază de date (PostgreSQL) pentru a stoca informațiile despre utilizatori și dispozitive.

Prezentare generală la nivel înalt:

- **Autentificarea utilizatorilor:** Utilizatorii se conectează la sistem, iar sistemul le validează credențialele și îi redirecționează în funcție de rol.
- **Controlul accesului bazat pe roluri:** Funcționalitățile sunt restricționate în funcție de rolurile utilizatorilor.
- **Comunicația microserviciilor:** Microserviciile comunică prin API-uri REST, permițând operațiuni precum asocierea utilizator-dispozitiv.



3. Diagrama de Deployment UML



4. Detalii de Implementare

Tehnologii Utilizate

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

Docker

Proiectul utilizează Docker pentru a izola și gestiona serviciile necesare. Fiecare serviciu este implementat într-un container Docker, care asigură portabilitate și consistență în mediul de dezvoltare și producție. Iată lista serviciilor Docker utilizate:

1. **Postgres pentru utilizatori:** Un container care rulează o bază de date PostgreSQL pentru microserviciul de gestionare a utilizatorilor.
2. **Postgres pentru dispozitive:** Un alt container PostgreSQL pentru microserviciul de gestionare a dispozitivelor.
3. **Microserviciul de gestionare a utilizatorilor:** Containerele care rulează aplicațiile Spring Boot pentru gestionarea utilizatorilor.
4. **Microserviciul de gestionare a dispozitivelor:** Containerele care rulează aplicațiile Spring Boot pentru gestionarea dispozitivelor.
5. **Frontend:** Containerul care rulează aplicația Angular.

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

Acest sistem de management al energiei oferă o soluție eficientă pentru gestionarea utilizatorilor și a dispozitivelor, integrând tehnologii moderne precum

Spring Boot, Angular și Docker. Prin implementarea Spring Security, sistemul asigură un nivel înalt de securitate, protejând datele și resursele aplicației.