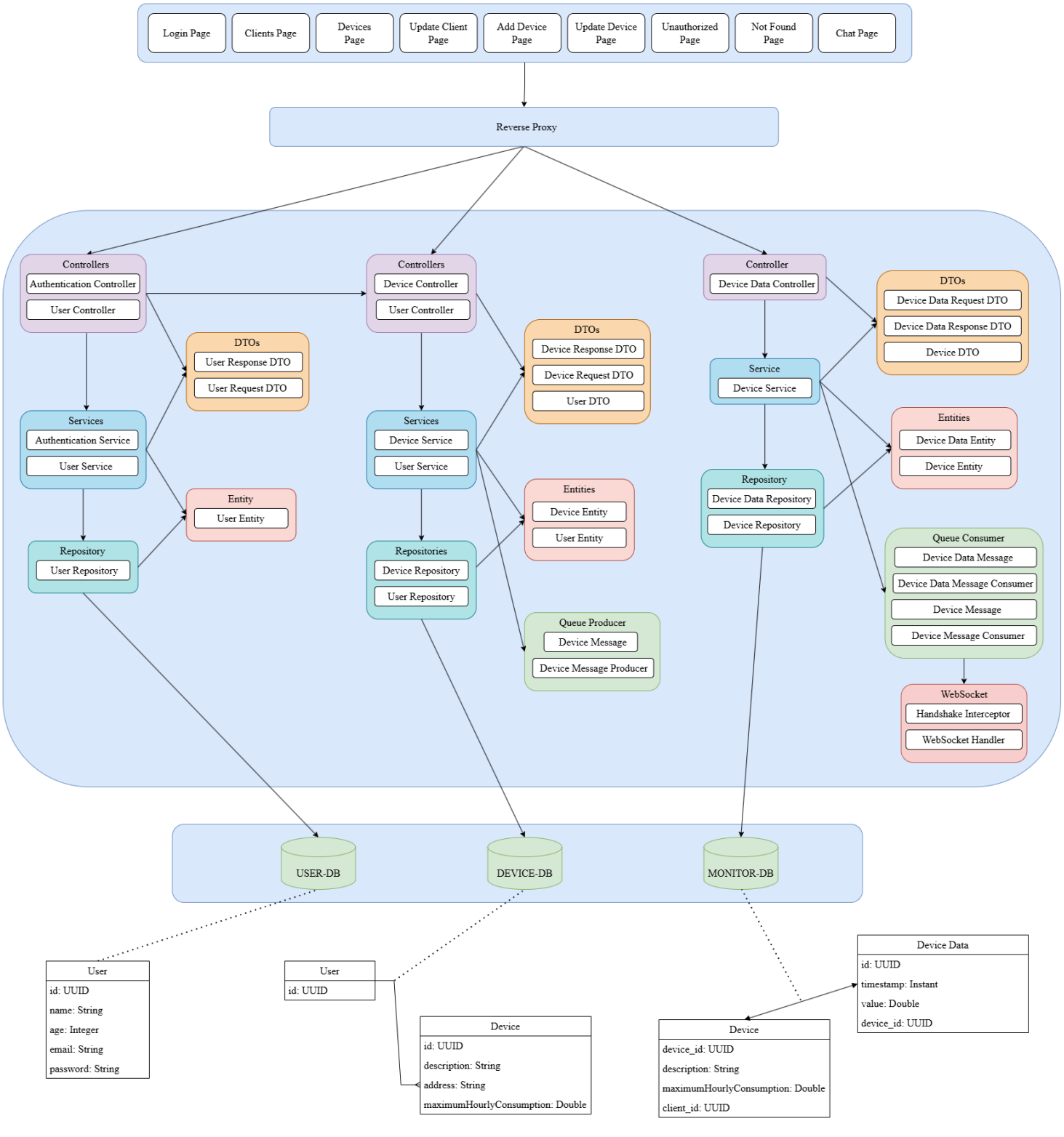
SISTEME DISTRIBUITE

TEMA 3

WEB SOCKETS AND SECURITY

I. ARHITECTURA CONCEPTUALĂ A SISTEMULUI DISTRIBUIT



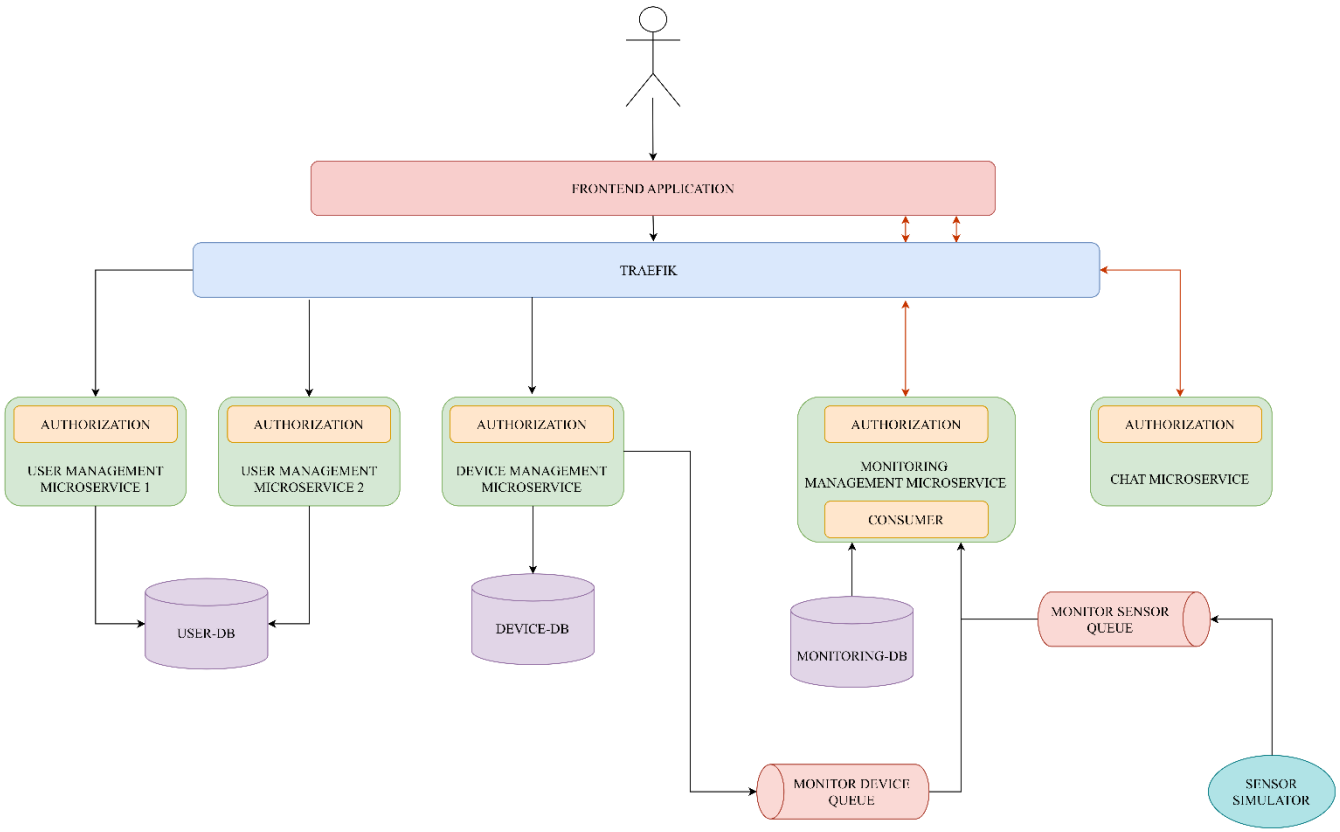
Arhitectura conceptuală a sistemului distribuit implementat este o arhitectură bazată pe trei nivele, care comunică între ele prin intermediul API-urilor.

Diagrama prezintă diferitele componente ale sistemului, fiecare având roluri diferite și aplicații specializate.

- Presentation Layer: Acest nivel este responsabil de interacțiunea cu utilizatorul și afișarea interfeței de utilizare. Include aplicația Frontend cu paginile de autentificare, chat, pagina principală pentru rolul clientului și cel al administratorului, cu opțiunea acestuia de navigare pe pagini diferite pentru înregistrarea unui dispozitiv asociat clientului selectat și actualizarea datelor clienților sau dispozitivelor.
- Application Layer: Acest nivel încorporează logica operațiunilor sistemului. Include aplicații pentru gestionarea utilizatorilor și dispozitivelor, precum și a datelor primite de la simulatorul de senzori.
- Data Layer: Acest nivel este responsabil de stocarea datelor. Include bazele de date pentru utilizatori, dispozitive si partea de monitorizare.

Nivelul Reverse Proxy se găsește între dispozitivele client și servere backend. Interceptează cererile de la clienți si le direcționează către microserviciul corespunzător.

Fiecare microserviciu are propria bază de date. Cel destinat dispozitivelor are nevoie de informații din partea celui destinat utilizatorilor pentru a păstra legătura între clienți și instrumentele înregistrate. Pentru a menține datele sincronizate s-a creat un tabel cu ID-urile utilizatorilor. Prin expunerea API-urilor, acesta este actualizat în momente cheie, cum ar fi adăugarea sau ștergerea unui client. Cel destinat monitorizării are nevoie de informații despre dispozitivele înregistrate pentru a le putea monitoriza consumul de energie. Prin folosirea unui topic rabbitmq, se introduc și se modifică datele necesare. Informațiile din partea simulatoruli de senzori sunt transmise microserviciului destinat monitorizării prin intermediul unui queue rabbitmq.

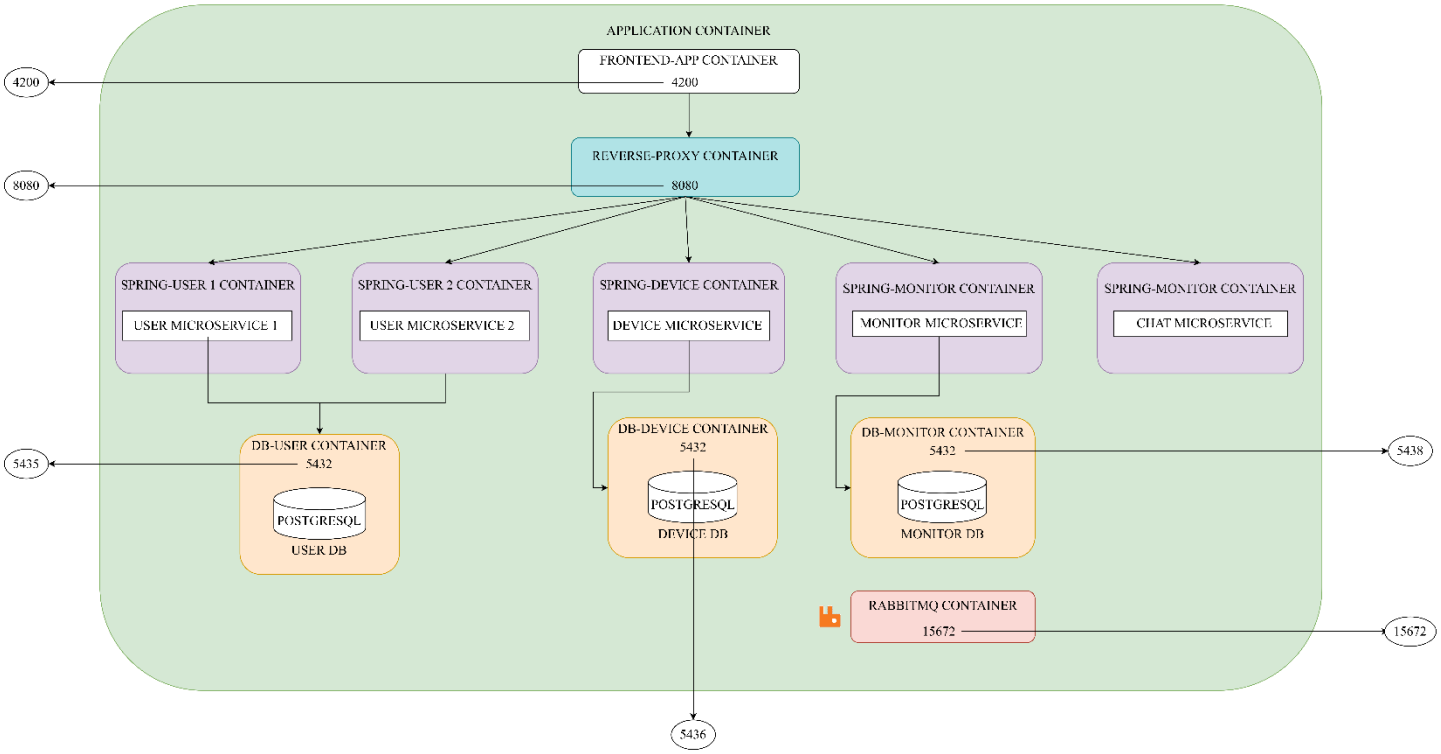


Interacțiunea utilizatorului cu sistemul funcționează în următorul mod:

- Utilizatorul accesează aplicația prin intermediul unei pagini web și interacționează cu nivelul de prezentare.
- Aplicația Frontend trimite cereri prin Traefik către microserviciul corespunzător, în funcție de acțiunile inițiate de utilizator. Aceste cereri pot ajunge la microserviciile de management al utilizatorilor, managementul dispozitivelor, managementul monitorizării sau chat, în funcție de tipul de operațiune solicitată.

- Nivelul de business procesează cererea și actualizează baza de date și nivelul de prezentare, astfel încât utilizatorul să vadă rezultatele acțiunii sale.
- Microserviciul de monitorizare colectează date din cozi, alimentate de un simulator de senzori și gestionate de un consumator, pentru a actualiza informațiile despre monitorizarea dispozitivelor în baza de date. De asemenea, comunică cu nivelul de prezentare prin intermediul unui WebSocket pentru a afișa o alarmă atunci când dispozitivul unui utilizator consumă mai multă energie decât cea înregistrată.
- Microserviciul de chat permite schimbul de mesaje între administrator și clienți. Mesajele nu sunt păstrate după închiderea ferestrei de comunicare. Acesta comunică cu nivelul de prezentare prin intermediul unui WebSocket pentru a afișa mesajele și statusurile lor.
- Microserviciile din cadrul sistemului utilizează autorizarea prin validarea unui token JWT. Acest token este stocat în cookie după ce autentificarea este efectuată cu succes.

II. DIAGRAMA UML DE DEPLOYMENT



Fiecare componentă din cadrul sistemului distribuit este reprezentată sub forma unui serviciu și este izolată într-un container separat. Microserviciul destinat utilizatorilor a fost replicat de două ori cu scopul de a introduce conceptul de load balancing în cadrul sistemului.

Microserviciile de utilizatori, dispozitive, monitorizare și chat sunt dezvoltate din imagini Docker personalizate, descrise în fișiere Dockerfile corespunzătoare. Fiecare serviciu are propriile variabile de mediu pentru conectarea la baza de date și configurarea interacțiunilor cu alte microservicii.

Serviciile de gestionare a datelor sunt configurate cu variabile de mediu care includ numele bazelor de date, utilizatorii, parolele și alte detalii relevante. Pentru persistența datelor, aceste servicii utilizează volume Docker asociate fiecărui container de bază de date.

Aplicația client este reprezentată de serviciul Frontend, care comunică cu celelalte servicii prin intermediul unei configurații de reverse-proxy. Acest proxy direcționează cererile către microserviciile corespunzătoare folosind porturile specifice fiecărui serviciu.

Toate aceste containere sunt integrate într-o rețea Docker externă denumită app_net pentru a asigura comunicarea între servicii.