

UNIVERSITATEA TEHNICĂ CLUJ-NAPOCA
FACULTATEA: Automatică și Calculatoare
SPECIALIZAREA: Tehnologia Informației
DISCIPLINA: Sisteme Distribuite
PROIECT: Energy Management System
Assignment 3

Îndrumător Laborator:
Dan Mitrea

Student:
Szikszai Csaba

Contents

Arhitectura Conceptuală a Sistemului Distribuit:	3
UML Deployment diagram	5

Arhitectura Conceptuală a Sistemului Distribuit:

User Interface:

Aplicația Angular servește ca interfață pentru Energy Management System. Oferă o platformă pentru utilizatori să interacționeze cu sistemul, în funcție de rolurile lor respective. Aplicația Angular comunică cu microserviciile pentru a efectua operațiuni CRUD asupra conturilor de utilizator, dispozitivelor și a asocierea acestora.

Microservicii:

Microserviciul de Management al Utilizatorilor:

- Responsabil pentru gestionarea operațiunilor legate de utilizatori.
- Expune endpoint-uri RESTful pentru crearea, citirea, actualizarea și ștergerea conturilor de utilizator.
- Comunică cu Baza de Date a Utilizatorilor.

Microserviciul de Management al Dispozitivelor:

- Gestionează operațiunile legate de dispozitivele inteligente de măsurare a energiei.
- Oferă endpoint-uri pentru operațiuni CRUD asupra dispozitivelor.
- Interacționează cu Baza de Date a Dispozitivelor.

Microserviciul de Monitorizare și Comunicare:

- Furnizează notificări în timp real utilizatorilor și dispozitivelor.
 - Utilizează WebSocket pentru comunicare instantanee.
 - Interacționează cu Microserviciul de Management al Dispozitivelor prin cozi RABBITMQ.
- Comunică cu o aplicație de simulare prin intermediul unei cozi de mesaje.

Baze de Date:

Baza de Date a Utilizatorilor:

- Conține informații despre utilizatori, inclusiv ID-ul, numele și rolul (admin/client).
- Utilizată de către Microserviciul de Management al Utilizatorilor.

Baza de Date a Dispozitivelor:

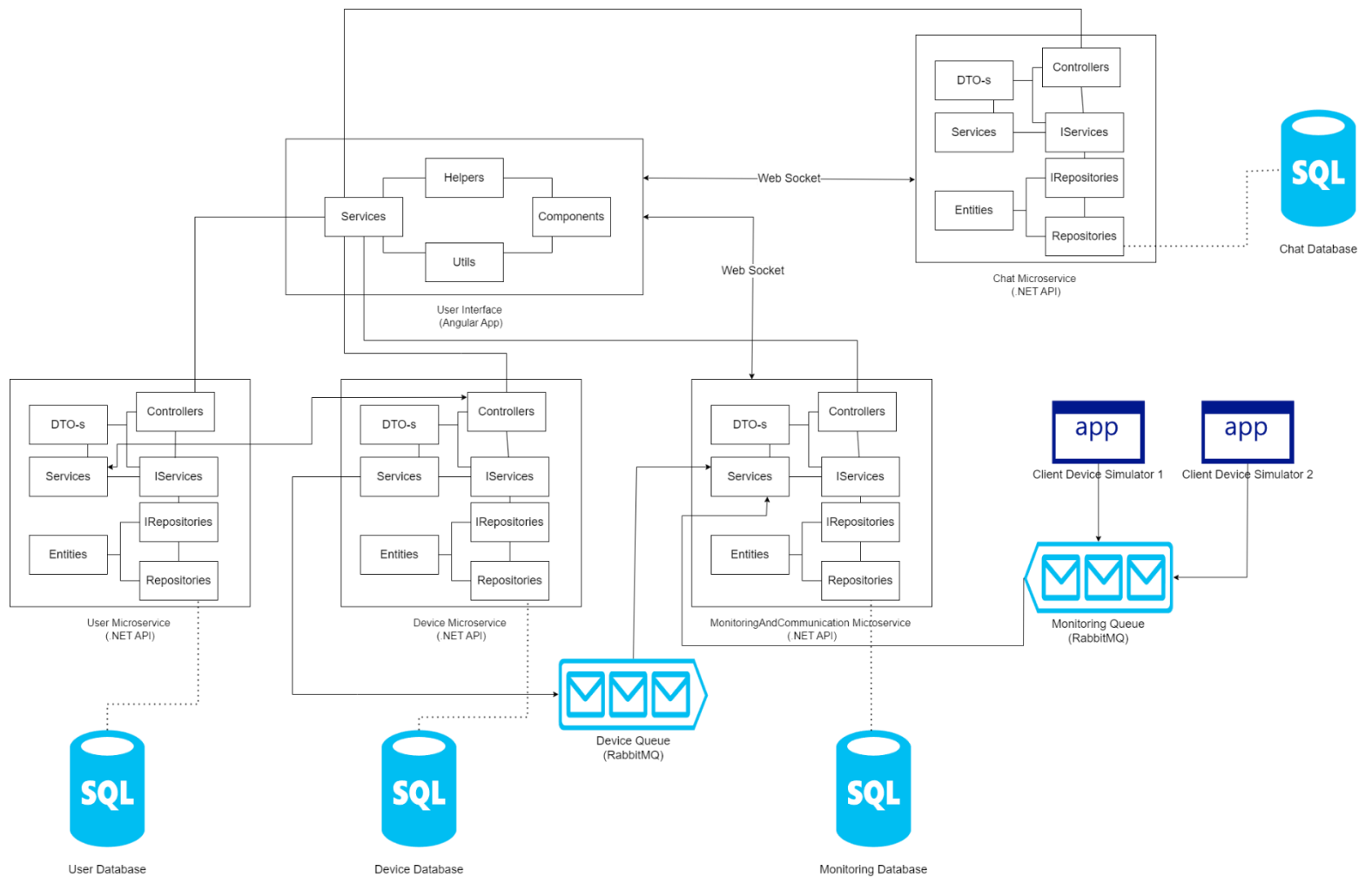
- Stochează detalii despre dispozitivele inteligente de măsurare a energiei, inclusiv ID-ul, descrierea, adresa și consumul maxim de energie pe oră.
- Accesată de către Microserviciul de Management al Dispozitivelor.

Baza de Date a Monitoring:

- Conține informații despre dispozitive și date de monitoring despre dispozitive

Containere Docker:

Aplicația Angular, împreună cu microserviciile de Management al Utilizatorilor, Dispozitivelor și Monitorizărilor și bazele de date și cozile de RabbitMQ sunt containerizate folosind Docker. Acest lucru asigură implementarea consecventă și reproductibilă în diferite medii.



UML Deployment diagram

Container 1: Server SQL (User database)

- Image: Server SQL
- Port: 1433
- Acest container găzduiește Server SQL pentru Baza de Date de User.

Container 2: Server SQL (Device database)

- Image: Server SQL
- Port: 1434
- Acest container găzduiește Server SQL pentru Baza de Date de Device.

Container 3: Microserviciu User

- Image: Microserviciu User
- Port: 5075
- Acest container rulează Microserviciul User folosind serverul Kestrel.

Container 4: Microserviciu Device

- Image: Microserviciu Device
- Port expus: 5121
- Acest container rulează Microserviciul Device folosind serverul Kestrel.

Container 5: Server NGINX cu Aplicație Front-end

Image: NGINX

Port expus: 4200

Acest container servește ca server NGINX și găzduiește aplicația front-end.

Container 6: Coadă de Monitoring (RabbitMQ)

- Imagine: RabbitMq
- Port expus: 5672
- Acest container găzduiește o coadă de Monitoring pentru a gestiona evenimente legate de monitorizare.

Container 7: Coadă de Device (RabbitMQ)

- Imagine: RabbitMq
- Port expus: 5673
- Acest container găzduiește o coadă de Device pentru a gestiona evenimente legate de dispozitive.

Container 8: Microserviciu MonitoringAndCommunication

- Imagine: Microserviciu MonitoringAndCommunication
- Port expus: 5173
- Acest container rulează Microserviciul MonitoringAndCommunication.

Interacțiuni și Comunicare

Microserviciu User <-> Microserviciu Device

- Microserviciul User comunică cu Microserviciul Device folosind numele lor de serviciu respective ('user-microservice' și 'device-microservice'). Acest lucru se poate face prin HTTP request.

Server NGINX <-> Microservicii

- Serverul NGINX face request-uri către microservicii folosind 'localhost:port', unde 'port' corespunde porturilor expuse ale microserviciilor respective (5075 pentru Microserviciul User și 5121 pentru Microserviciul Device).

Microserviciu MonitoringAndCommunication<-> Coadă de Device <-> Microserviciu Device

- Microserviciul MonitoringAndCommunication interacționează cu Coadă de Device pentru a primi mesaje.
- Microserviciul Device interacționează cu Coadă de Device pentru a trimite mesaje.

