

DISTRIBUTED SYSTEMS

Assignment 3

Web sockets and security

1 Microservicii

User Service

User service este un microserviciu care este responsabil atat de securitatea aplicatiei, deoarece are un layer de spring security, cat si de conexiunea clientilor si mentionarea datelor despre conturile acestora. Există toate operatiile CRUD care se efectueaza pe useri. De asemenea, acesta are o legatura stransa si comunica cu serviciul de devices prin intermediul unor http calls care sunt realizate cu ajutorul unui Feign Client. Aceasta comunicare este necesara datorita legaturii dintre

un client si un device. Fiecare device avand un owner si anume un client. Aceasta legatura este de fapt o relatie de one to many intre cele doua baze de date ale celor 2 servicii mentionate mai sus. Astfel un device poate fi asignat unui singur user, in timp ce un user poate detine mai multe device-uri.

Device Service

Device Service este un serviciu care se ocupa cu stocarea, adaugarea, modificarea si stergerea device-urilor. Pe langa legatura de one to many dintre baza de datea devices si cea a users, mai exista inca o legatura dintre baza de date a deviceurilor si cea a serviciu-lui de monitorizare. Legatura existenta este realizata prin intermediul unui message broker (RabbitMQ) si este folosita pentru a mentine datele sincronizate intre cele 2 baze de date ar trebui sa reflecte aceleasi date.

Monitoring Service

Monitoring Service este un serviciu care se ocupa cu monitorizarea consumului de energie pentru fiecare device assignat unui user. Acesta are o baza de date formata din doua tabele:

- Tabelul de sincronizare - care preia mereu datele trimise prin rabbitmq de la serviciul de devices si updateaza orice schimbare care a avut loc la nivel de device. Acest tabel contine device guid, user guid si max consumption.
- Tabelul de energy consumption - preia datele trimise de la fiecare dintre senzorii deviceurilor care trimit un consum la fiecare 10 minute. Acest tabel contine device guid, consumption si timestampul la care a fost citite datele de pe senzor.

Nevoia de a avea o baza de date care sa fie in permanenta sincronizata cu cea din serviciul de deviceuri este din cauza faptului ca un senzor trimite doar un device guid iar mai apoi serviciul de monitorizare trebuie sa faca legatura dintre device si userul care il detine pentru a putea afisa mai apoi informatiile in timp real in aplicatie doar pe pagina userului care detine acest device. De asemenea serviciul de monitorizare verifica in permanenta daca maximul admis de energie consumata a fost depasit si alerteaza userul in aplicatie prin afisarea unui semn de pericol deasupra acelui device. Conectiunea dintre serviciu si aplicatia web este facuta cu ajutorul web socketurilor.

Chat Service

Chat Service este un serviciu care se ocupa cu centralizarea, trimiterea si primirea mesajelor in intermediul aplicatiei dintre utilizatorii, care au nevoie de asis-

tenta, si administratorii care se ocupa cu service suport si care pot sa raspunda la intrebari. Aceste live chat este realizat prin intermediul unui serviciu care comunica atat cu clientul adminului si cat si cu cel al userului prin intermediul websocketurilor. Astfel un user se conecteaza la un canal special desemnat pentru el care are o structura de tipul : `"/chat/userGuid/message"` unde `userGuid` este partea care asigura unicitatea canalului de comunicatie dintre microserviciu si userul respectiv. Un admin, spre deosebire de user, se conecteaza la un canal comun prin care se transmit toate mesajele de la toti userii spre pagina de admini, unde oricare dintre admini conectati pot raspunde la intrebari. Asadar acest microserviciu este o cale de a crea o conexiune bilaterala intre user si admini de tipul one to many.

Pentru implementarea de `seen` si `typing` am folosit aceleasi websocketuri de trimitere a mesajelor. La baza `seen` si `typing` sunt niste obiecte mesaj, dar care nu contin un mesaj de tip text, ci sunt trimise ca si tipu de `seen` si `typing`. In functie de tipul care este trimis de la un client la chat service. Acesta va redirectiona prin calea de websocket mesajul de `typing` sau `seen` la clientul cu care adminul discuta. In functie de tipul de mesaj ce a ajuns de la microserviciul de chat in partea de front end se vad niste verificari si se va afisa mesajul corespunzator in chat. Daca un admin sau client deschide chatul si exista mesaje se va trimite automat un mesaj de tipul `seen` la celalalt participant la chat. Daca oricare din ei trimite un mesaj nou `seen` este eliminat si sa va astepta apasarea pe textfield pentru a retrimite un mesaj `seen`. Pentru partea de `typing` de fiecare data cand un participant apasa pe text field apare mesajul de `typing`... si cand acesta nu mai are focus pe textfield se trimite un mesaj de `stopTyping`.

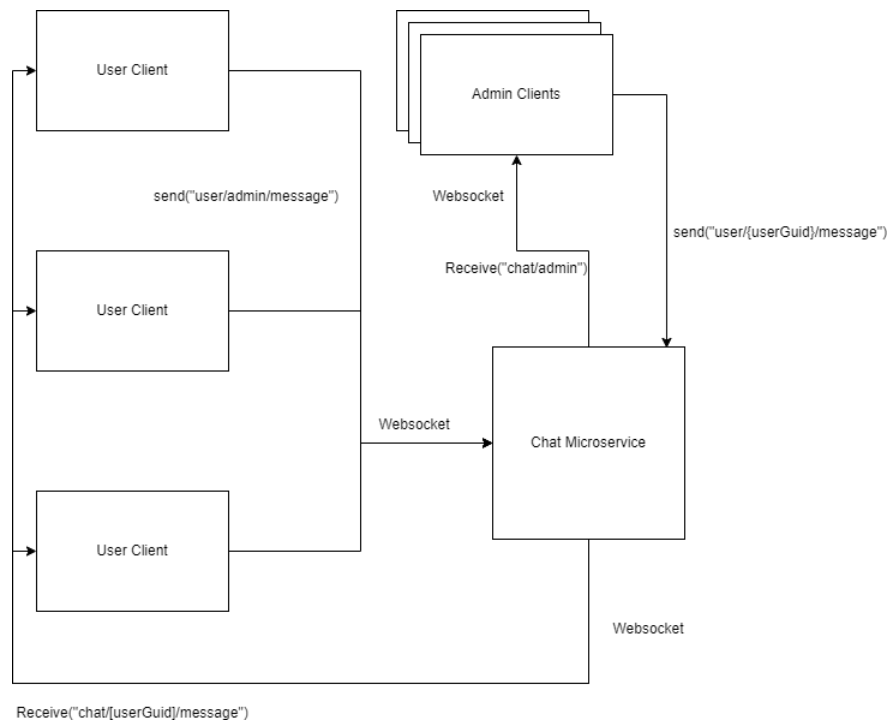


Figure 1: Arthitectura Chat Service

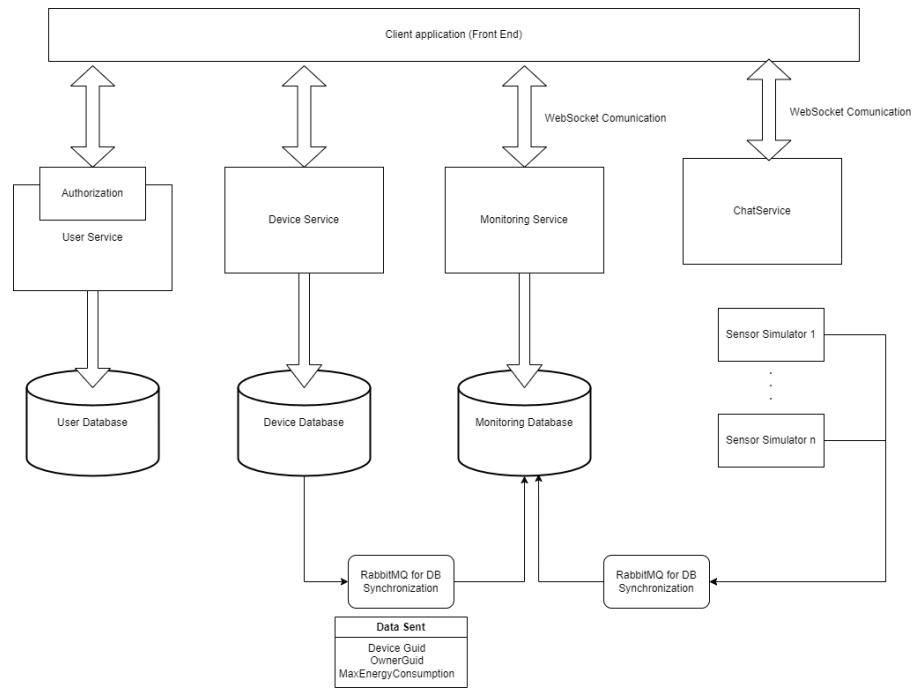


Figure 2: Arhitectură Bazată pe Microservicii

Diagram de deployment

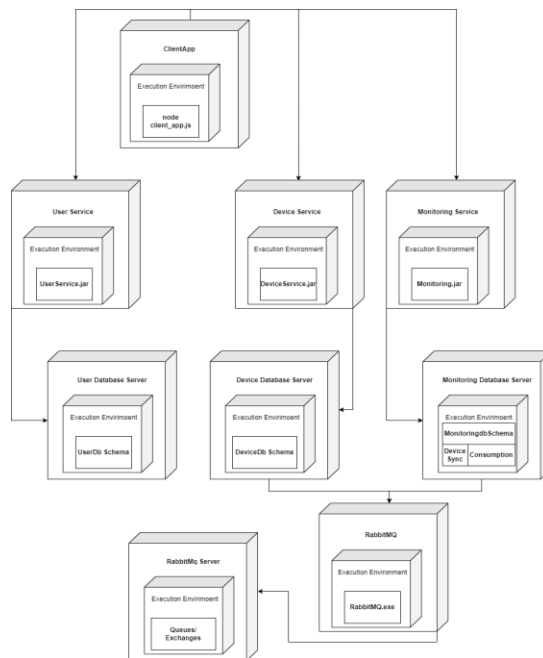


Figure 3: Deployment

2 Autentificare cu JWT

JSON Web Token (JWT) este un standard deschis pentru transmiterea de informații între părți într-un format compact și sigur. Aceste tokens sunt utilizate pentru autentificare și autorizare în aplicațiile web și serviciile web.

Pentru aceasta parte am folosit jwt authentication pentru multiple servicii pe care aplicația le folosește. Sistemul funcționează în felul următor: Când un user se conectează, se va apela serviciul de users care va valida usernameul și

parola. Daca acestea sunt corecte, cu ajutorul unui jwt secret, pe care toate microserviciile il detin, si care este salvat in `application.properties`, se va genera un token care va ramane valabil pentru un anumit timp. Dupa ce acest token a fost generat la fiecare apel catre alte microservicii cum ar fi `device service`, se va face o filtrare si o verificare a validitatii tokenului care a fost transmis in header. Astfel daca un call este realizat, dar tokenul nu este valid se va primi un raspuns automat de 403(`Access Forbidden`).

Autentificarea cu JWT se desfășoară în următorii pași:

1. Autentificare
2. Generarea Tokenului
3. Transmiterea Tokenului
4. Verificarea Tokenului