

# SISTEME DISTRIBUITE

Tema 1 - Energy Utility Platform

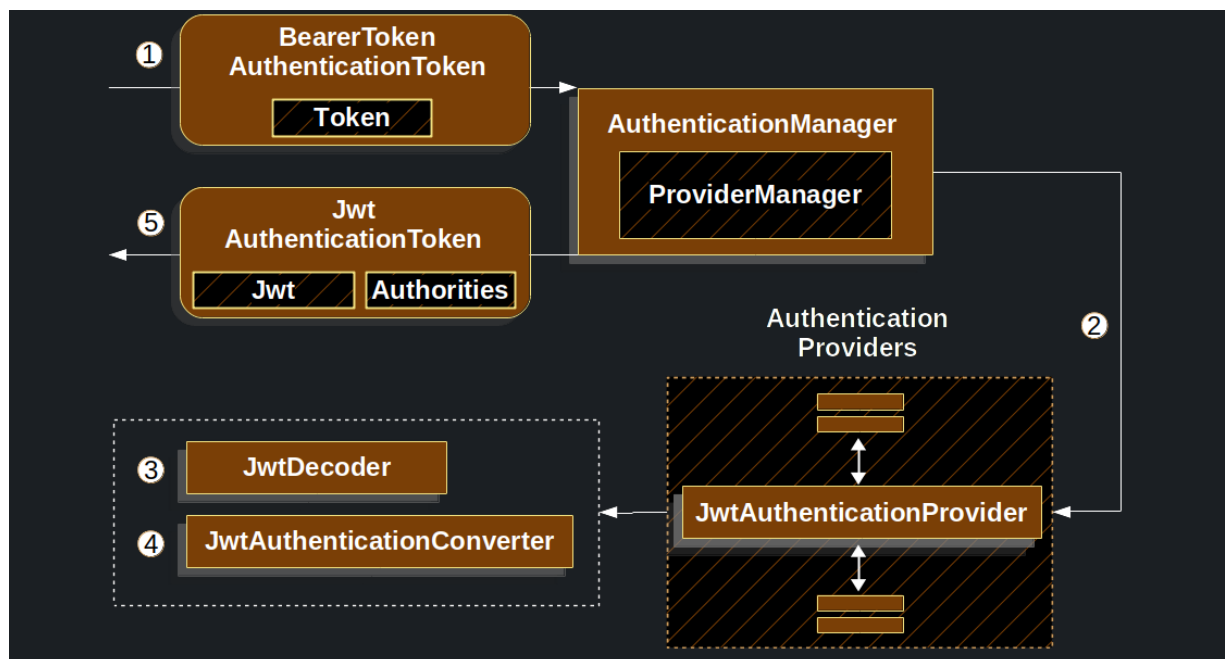
Mariciuc Andrei-Alexandru

# Conceptual architecture of the online platform

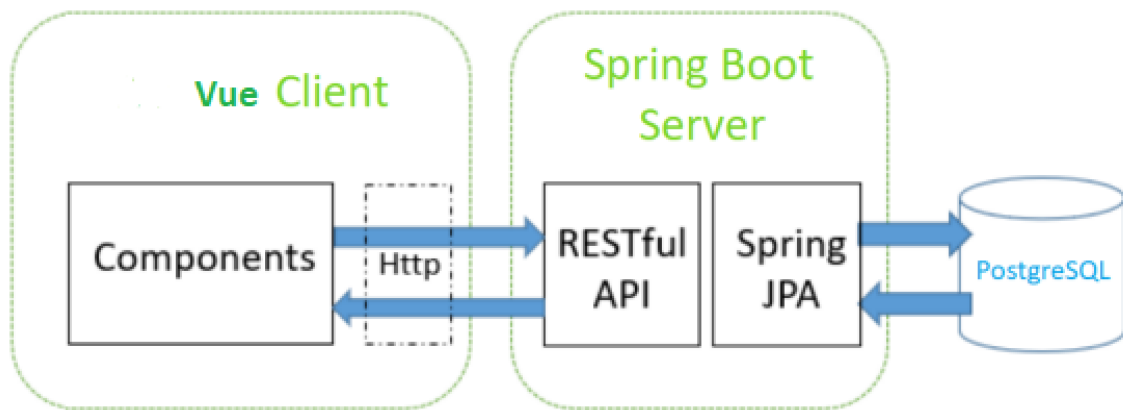
Aplicația creată este destinată gestiunii unor dispozitive consumatoare de energie. Dacă ești utilizator o să poți să-ți vezi dispozitivele, adresa la care se afla acestea și consumul de energie, iar dacă ești administrator o să poți șterge/actualiza sau crea dispozitive și legăturile către diverși utilizatori.

Aplicația este structurată în două parti, una care se ocupă de gestiunea internă a datelor, iar una care se ocupa in mare parte de afisarea lor.

Primul modul a fost implementat utilizand Spring Boot, și este modulul principal al aplicației el fiind responsabil de expunerea datelor și de forma acestora. Persistenta a fost asigurata de PostgreSQL, o baza de date relationala si de JPA (Java Persistence API) care este construit peste Hibernate un ORM reductabil in lumea Java. Modulul responsabil de afisarea datelor care aveau sa fie deservite de serverul pe care rula Spring Boot este bazat pe Vue, un framework care ajuta la implementarea aplicatiilor single-page. Pentru o comunicare securizata s-a folosit JWT (Json Web Token), o metoda bazata pe cheie publica, cheie privata pentru securizarea login-ului. Mai exact, acest token are o semnatura care va fi incriptata folosind cheia privata, iar apoi cand token-ul va veni de pe frontend va fi decriptat cu o cheie publica corespunzatoare si mai apoi validat. Un astfel de token, are posibilitatea adaugarii anumite claim-uri, cum ar fi numele utilizatorului, rolul etc. Aceste claim-uri vor ajuta la validarea datelor venite de la front-end.



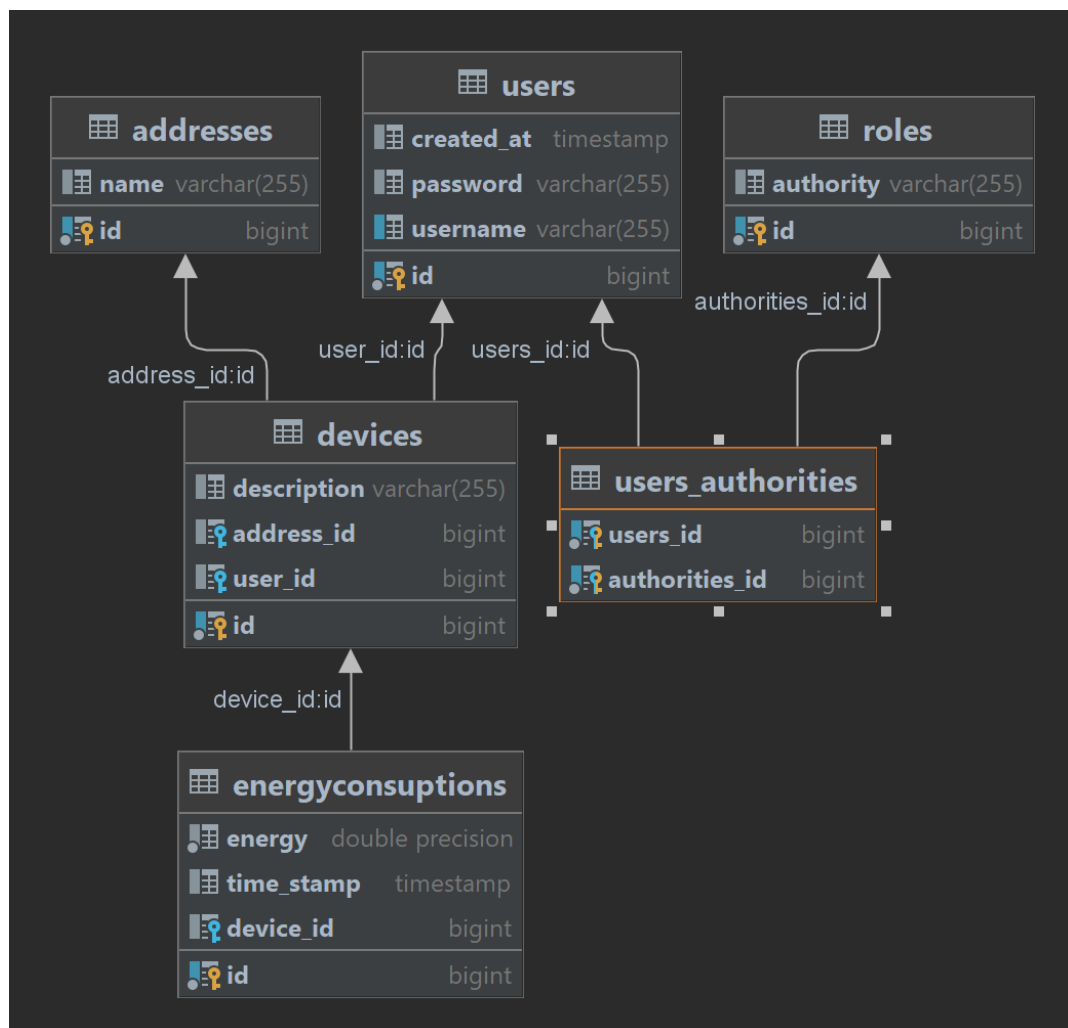
Metoda de lucru a securității în aplicație



Arhitectura conceptuala

## DB DESIGN

---



- fiecare utilizator are mai multe roluri, si un rol poate sa aiba mai multi utilizatori - many to many
- fiecare utilizator are mai multe device-uri - one to many
- fiecare device are mai multe tuple de consum de energie - one to many
- la o adresa putem avea mai multe deviceuri - many to one

## UML Deployment diagram

---

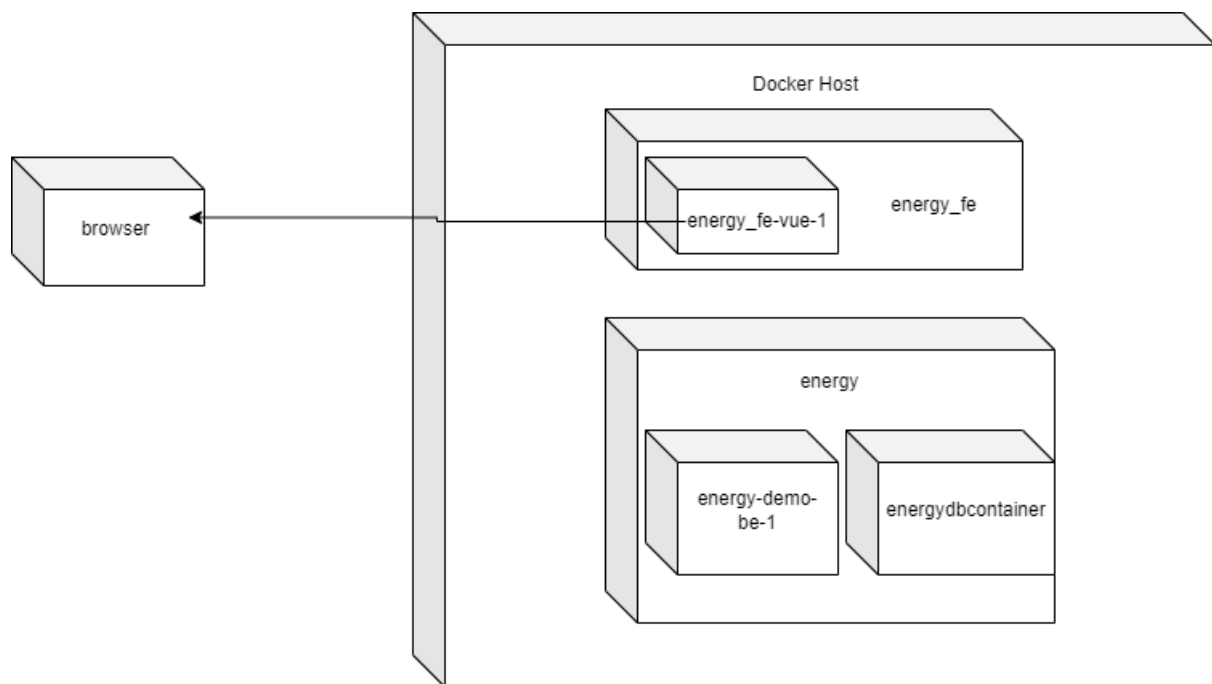


Diagrama de deploy locală

## LECTII INVATATE

---

- Ar trebui acordata o atentie mai mare modului in care se face modelarea la nivel de ORM, pentru ca o modelare gresita poate sa aduca multe schimbari in cod, modelele fiind piatra de capatai a intregii aplicatii.
- Ar trebui sa tin starile pe FE cat mai restranse. O stare mare, cu multe tab-uri nu este cea mai buna idee, pentru ca o schimbare cu influente mari poate fi greu de gestionat.