



# Documentație Assignment 1

## Online Energy Utility Platform

### 2022 – 2023

Nume și prenume Student:

Popescu Eduard-ionut

Grupa 30242



## 1. Arhitectura conceptuala a aplicatiei

Arhitectura aplicației trebuie să deservească specificul acesteia. Specificul aplicației date este o platformă online de monitorizare a energiei. Aceasta permite administratorului să atribuie dispozitive de consum cu clienți, iar clienții pot urmări detaliat consumul acestora.

Arhitectura proiectată este construită pe arhitectura REST prin crearea unui client și a unui server unde serverul are toate datele și clientul îi cere acestuia informații. Aceste informații se trimit sau se primesc prin folosirea protocolului HTTP pentru comunicare

Clientul aplicației este realizat prin intermediul framework-ului ReactJs, acest framework înglobează partea de JavaScript, HTML și CSS. Este un framework bazat pe componente pentru aplicații web ușor scalabile. Acesta permite acces cu ușurință la funcționalități precum routing, form management, comunicare client server. Aceasta parte de UI realizată prin intermediul tehnologiei prezentate mai sus ne ajută la expunerea funcționalităților prezente în controllerul din server pentru ca un utilizator să aibă un mediu mai ușor de folosit.

Serverul aplicației se ocupă cu logica aplicației unde regăsim partea de controllers prezentată mai sus. Arhitectura din server este bazată pe Layer Architecture deoarece avem o partitionare a fluxului pe multiple etaje de logica Controller -> Service -> Domain -> Repository -> DAO. În realizarea serverului am utilizat Spring-Boot care este un framework pentru Java, limbaj în care este scris serverul nostru, iar baza de date este de tipul relatională (PostgreSQL).

Beneficiile acestui tip de arhitectură sunt:

1. Există o dependență redusă deoarece funcția fiecărui strat este separată de celelalte straturi.
2. Testarea este mai ușoară datorită componentelor separate, fiecare componentă putând fi testată individual.

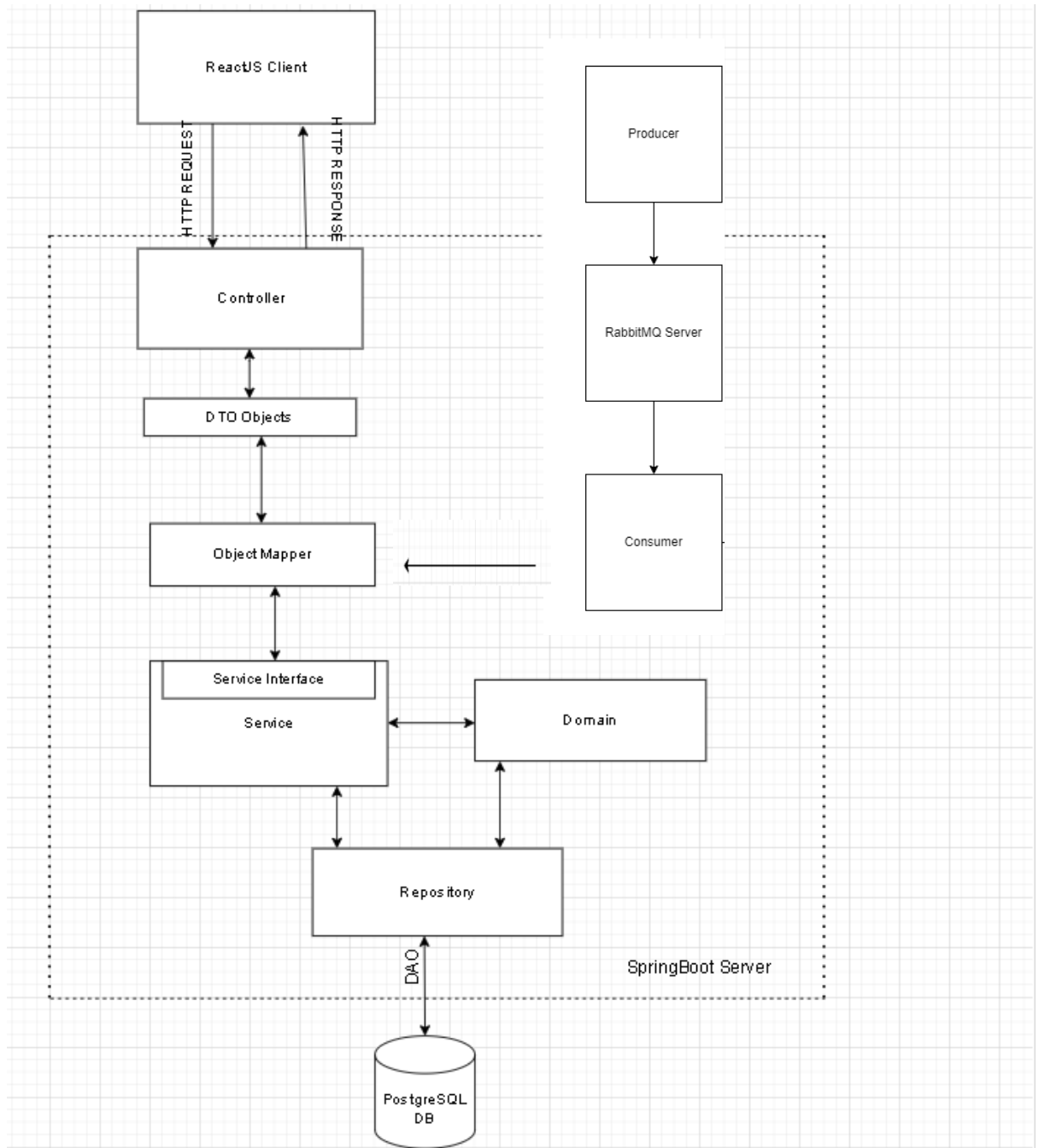
Totuși există și dezavantaje pentru această arhitectură:

1. Scalabilitatea este dificilă deoarece structura cadrului nu permite creșterea.
2. Pot fi greu de întreținut. O modificare într-un singur strat poate afecta întregul sistem deoarece funcționează ca o singură unitate. O posibilă îmbunătățire în această direcție este divizarea în microservices pentru cazul în care aplicație devine extrem de mare și greu de întreținut.

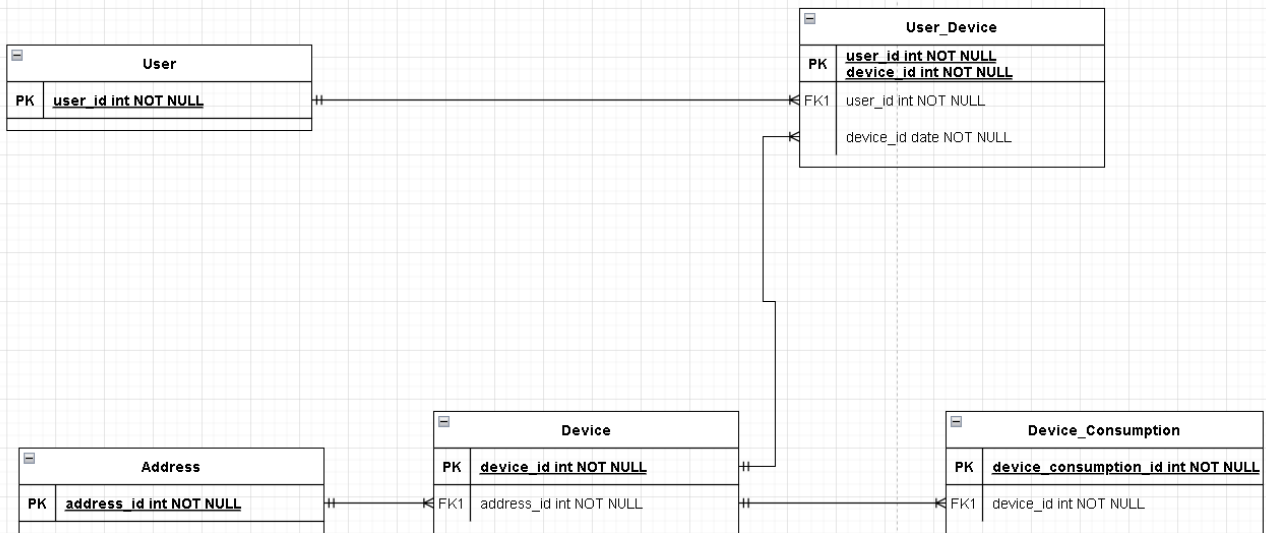


3. Există interdependență între straturi, deoarece un strat depinde de stratul de deasupra lui pentru a primi date.

**Assignment 2 :** Pentru acest assignment s-a solicitat implementarea unei cozi asincrone de mesagging pentru a realiza citirea de date de la un sensor. Impreuna cu implementarea unui producer de mesaje care transmite un mesaj la un interval fix de timp, s-a realizat si deployul containerelor care contin frontendul si backendul, aplicatia de producer a mesajelor fiind conectata la o coada asincrona in cloud usor de accesat



## 2. Database Diagram



### 3. Deployment diagram

