



# **Sisteme Informatice Distribuite**

## **Assignment 3**

### **Remote Procedure Call (RPC)**

### **Smart Home Appliance**

Student: Handaric Cosmin

Specialitate: Tehnologia Informatiei

Grupa: 30644

An universitar: 2021 – 2022

Asistent laborator: Alex Florea

## 1. Arhitectura conceptuala

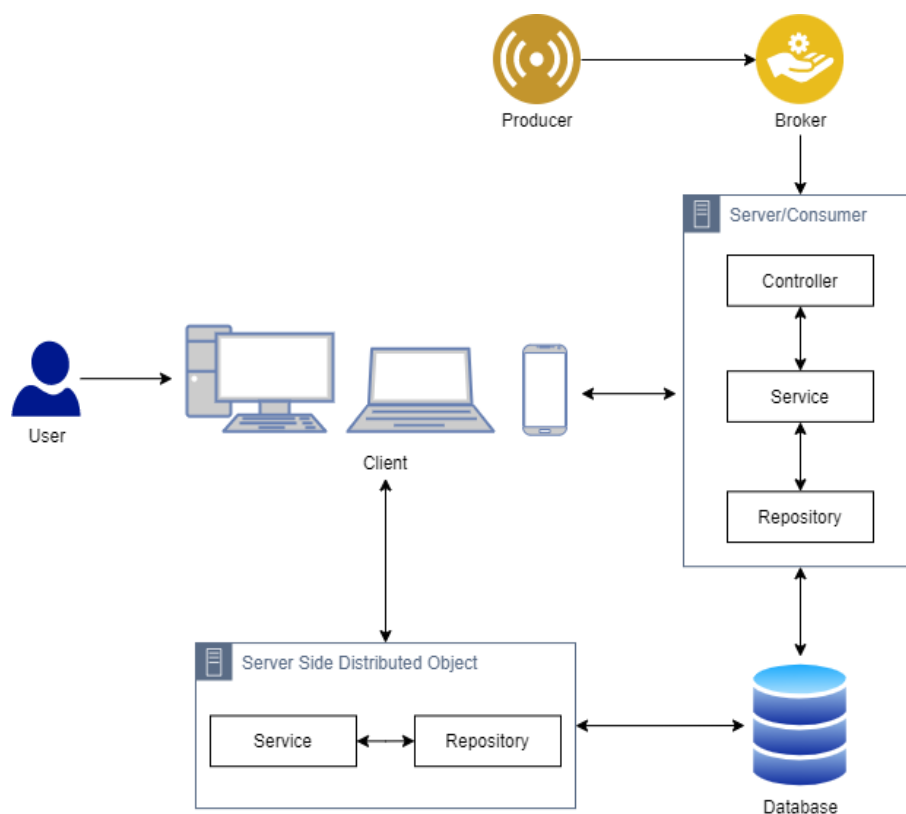


Figura 1.1

În primul rând, sistemul este dezvoltat pe baza unei arhitecturi de tip client-service, ce are ca și scop preluarea diferitelor acțiuni ale utilizatorului (componenta de client), urmând o prelucrare a acestora (componenta server), comunicarea realizându-se fie prin intermediul serviciilor REST, fie prin protocolul RPC.

Arhitectura mai surprinde și prezenta a trei module folosite pentru a genera anumite date și a le trimite către baza de date pentru a fi salvate. Aceste module fiind: **Producer**, **Broker** și **Consumer** (componenta server).

## 2. UML Deployment diagram

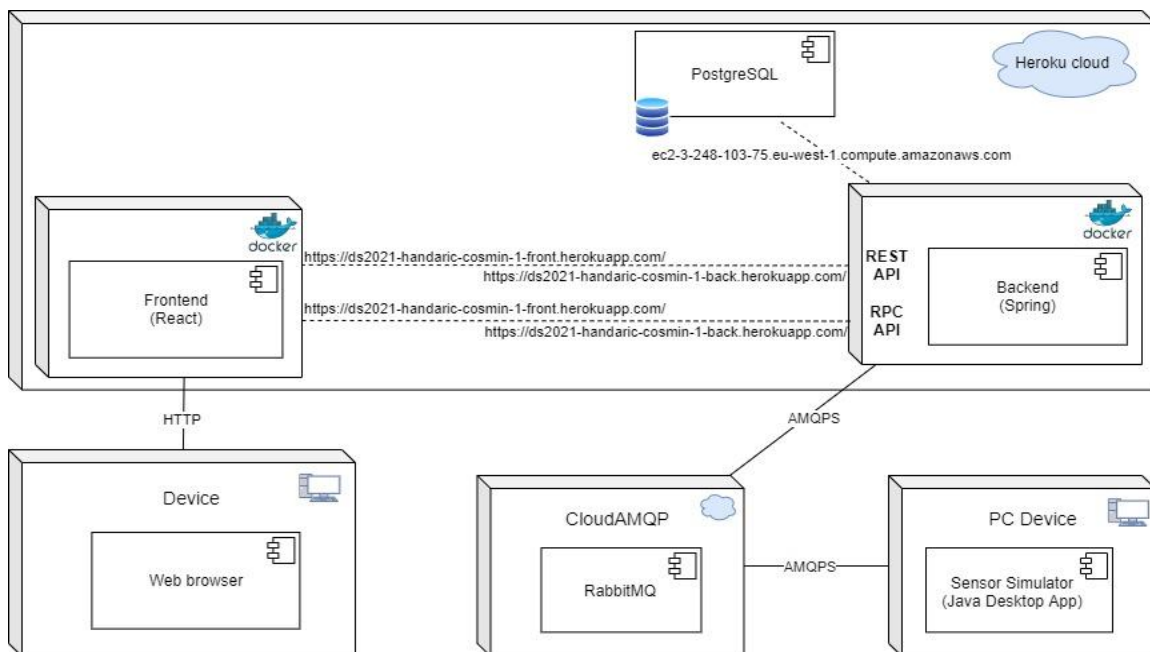


Figura 1.2

Pentru a realiza deploymentul sistemului s-au folosit serviciile oferite de platforma de cloud: *Heroku*. Au fost create doua aplicatii pentru deploy pe acesta platforma: una pentru componenta de server, a doua pentru componenta de client. Pe langa generarea aplicatiilor *Heroku* pentru deploy, pe partea de server s-a mai utilizat serviciul de *Heroku Postgres* pentru a accesa o baza de date aflata in cloud.

Atat aplicatia server, cat si aplicatia client au fost integrate in componente ce contin configuratiile minime ale unui sistem de operare si librariile si dependintele necesare pentru a fi rulate aplicatiile respective. Aceste componente se numesc containere si pentru a realiza acest proces s-a folosit platforma *Docker*.

Modul de comunicare dintre client si server se poate realiza prin 2 API-uri; unul configurat pentru serviciile REST, al doilea configurat pentru protocolul RPC (pentru aplicatia de monitorizare a consumului dispozitivelor unui client).

Partea de deployment a broker-ului a constat in folosirea cloud-ului *CloudAMQP*. Aceasta prezinta un set de servere unde se regasesc mai multe instante de *RabbitMQ*, software-ul responsabil de preluarea datelor de la producer si trimiterea acestora catre consumer.

Modulul de producer reprezinta o aplicatie Java Desktop locala.

## 3. Readme

### 3.1. Aplicatia web

Pentru a accesa aplicatia aflata pe cloud se poate accesa link-ul urmator: <https://ds2021-handaric-cosmin-2-front.herokuapp.com/>. Prima pagina va reprezenta pagina de autentificare, unde utilizatorul va trebuie sa isi introduca credentialele. In functie de rolul pe care il are (admin sau client) va fi redirectionat pe o pagina home.

Rute admin:

- */admin*: pagina principala a adminului
- */admin/clients*: vizualizarea clientilor
- */admin/clients/add*: aduagare clien
- */admin/clients/update/:id*: actualizare client
- */admin/clients/devices/:id*: vizualizarea dispozitivelor unui client
- */admin/devices*: vizualizare dispozitive
- */admin/devices/add*: adaugare dispozitiv
- */admin/devices/add/:id*: adugare dispozitv pentru un client specific
- */admin/devices/update/:id/:idDevice*: actualizare dispozitv
- */admin/my-profile*: profil admin
- */admin/change-password/:id*: schimbare parola
- */admin/sensor/export*: exportarea ID-urilor senzorilor intr-un fisier csv

Rute client:

- */client*: pagina principala a clientului
- */client/my-devices*: vizualizarea dispozitivelor clientului
- */client/my-devices/:id*: vizualizare detalii dispoziv
- */client/consumption*: vizualizare consum
- */client/notification*: vizualizarea mesajelor de notificare
- */client/my-app*: aplicația de vizualizare a consumului

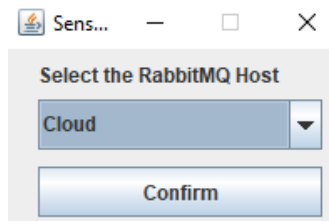
### 3.2. Simulator

Partea de simulare a functionalitatii senzorilor se realizeaza prin intermediul aplicatiei Desktop: **DS2021\_30644\_Handarc\_Cosmin\_2\_Producer.jar**.

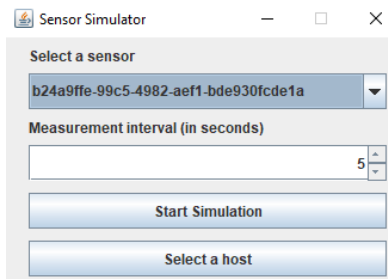
Datele de citire trebuie sa se afle in folderul **data**, iar formatul fisierelor trebuie sa fie urmatorul:

- *sensor.csv* – locul unde se afla toate ID-urile senzorilor (acesta se genereaza de catre admin din aplicatia web);
- *id\_sensor.csv* – aici se afla valorile masurate pentru un anumit senzor (*id\_sensor* reprezinta ID-ul senzorului).

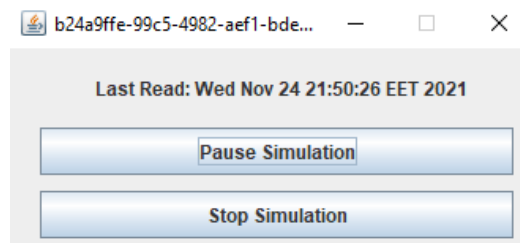
Aplicatia prezinta urmatoarea interfata:



Pas 1 – Alegerea broker-ului (Local – pentru development sau Cloud – pentru production)



Pas 2 – Alegerea senzorului si a intervalului de simulare (in secunde)



Pas 3 – Simulare masuratori pentru senzorul selectat anterior



### 3.2 Aplicatia de monitorizare a consumului

Aplicatia poate fi accesata de fiecare client pe ruta: */client/my-app* si este configurata in functie de device-urile acestuia.

Aplicatia prezinta trei functionalitati:

1. Vizualizarea unui graf ce prezinta consumul tuturor device-urilor in fiecare ora pe parcursul unui anumit numar de zile introdus de catre client
2. Vizualizarea baseline-ului clientului (consumul mediu pe saptamana a tuturor device-urilor)
3. Estimarea unui program optim de pornire a unui device (se selecteaza device-ul ce se doreste a fi pornit, apoi se regleaza programul de functionare, iar aplicatia va calcula ce mai buna ora de pornire a acestuia si va estima cat ar consuma acesta – se foloseste valoarea maxima ce poate fi atinsa de senzorul device-ului in estimare)