Distributed systems

Assignment 3

Remote Procedure Call (RPC) Smart Home Appliance

Pasc Andreea-Madalina
30441

1. Requirements:

Suppose that the clients have intelligent home appliances that can be controlled remotely using remote procedure call (RPC). Each such device can communicate with the server that will compute the time when the device will be started for an optimal energy consumption. Develop a client-side application (either a desktop application or a web application based on a JavaScript framework running from the browser) for the smart appliance associated to a client that:

i) gets the client hourly historical energy consumption over d days in the past ($Eclient\ d\ (h)$);

ii) gets the averaged energy consumption for the client over the past week (e.g. client baseline); $Baseline(h) = 1$ to $7\sum Eclient\ d\ (h)\ 7\ d{=}1\ ,\forall h \in \{1.\ .24\}$, $where\ Eclient\ d\ (h)\ is\ the\ client\ energy\ consumption\ for\ day\ d\ in\ the\ past\ and\ hour\ h\ from\ day\ d$

iii) allows the selection of a program with a duration in hours (select a duration D of a program);

iv) gets the best time to be started considering the baseline and the program duration to avoid energy peaks from the client (e.g. to minimize the maximum energy consumption for every hour of the day) $Compute\ ts\ ,te\ such\ that\ Min(Max(Baseline(h) + EDevice)),\forall h \in [ts\ ,te\ ],te = ts + D$

2. Implementation

a. Server:

The server is integrated in the deployed backend from Assignment1 and it was done using Hessian RMI communication. For this, I put 3 classes: an interface with the function getConsumptions () which returns all the measurements in a List<ConsumptionDTO> using the findAll() function from ConsumptionService.

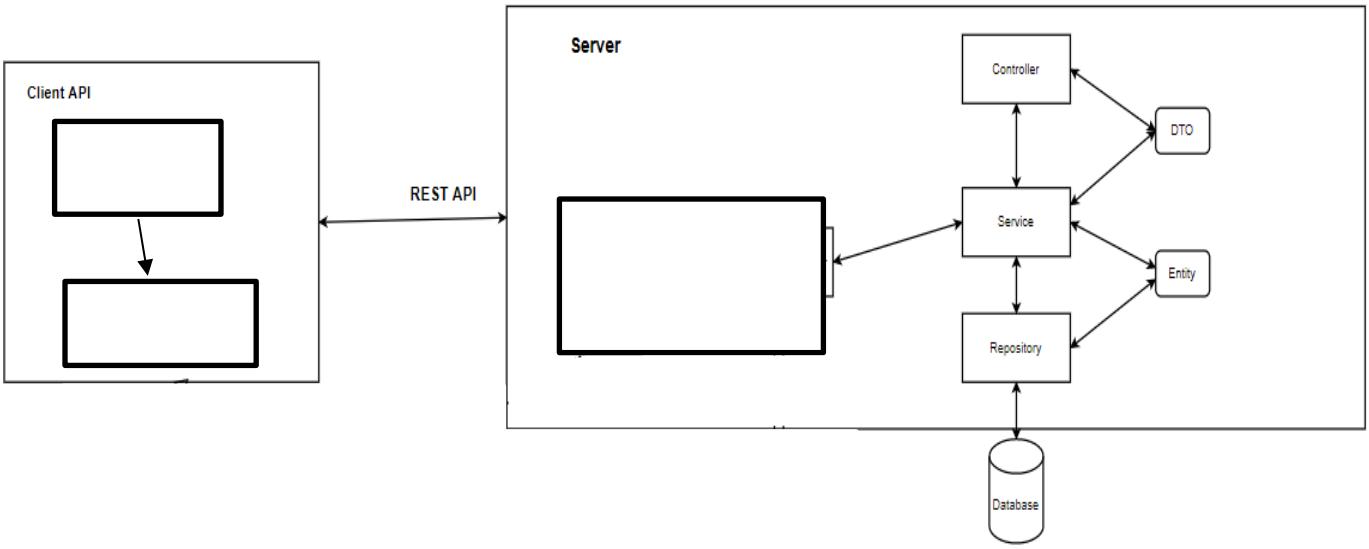
After that I implemented another class, a service class in order to implement the function from the interface. The hessian configuration was added in an additional Configuration class.Queue:

b. Client:

On the client side, using the path put in the configuration class and the interface with the getConsumptions() I extracted the values sent from the server in order to process them. I made a desktop app using Java Swing and JFrame with 5 different buttons:

- One button is meant to give the graph for all the historical consumptions
- One button is used to put the graph for the Baseline formula
- 3 additional buttons for the programs in which we get the start and end time for the graph:
    - One duration is of 3 hours
    - One duration is of 5 hours
    - One duration is of 8 hours

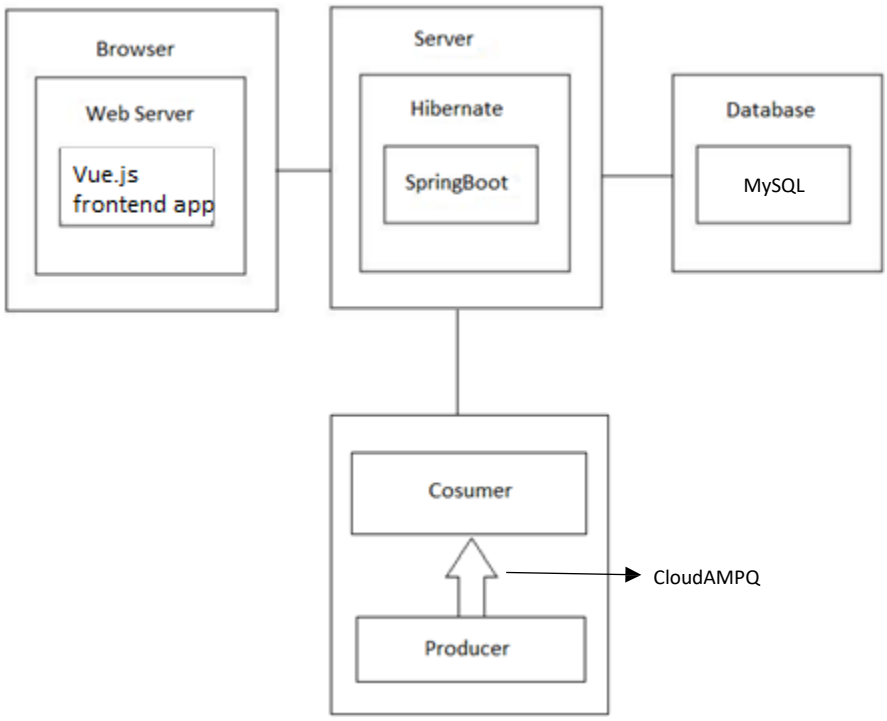
3. Conceptual architecture of the distributed system



4. UML Deployment diagram

The deployment of the backend started by creating a new app in Heroku, called “ds2020-30441- pasc-andreea-be”. After pushing the backend on gitlab, the name of the application from Heroku and the key were provided in the yml and Dockerfile. Also, in the variables there was created the connection with the deployed data base.

The GUI is a standalone application which is run locally from IntelliJ.


Browser
Web Server
Vue.js
frontend app
Server
Hibernate
SpringBoot
Database
MySQL
Cosumer
CloudAMPQ
Producer