DOCUMENTATATION

ASSIGNMENT 3: ENERGY MANAGEMENT SERVICE

STUDENT NAME: PELLE ANDREI
GROUP: 30441

TABLE OF CONTENTS

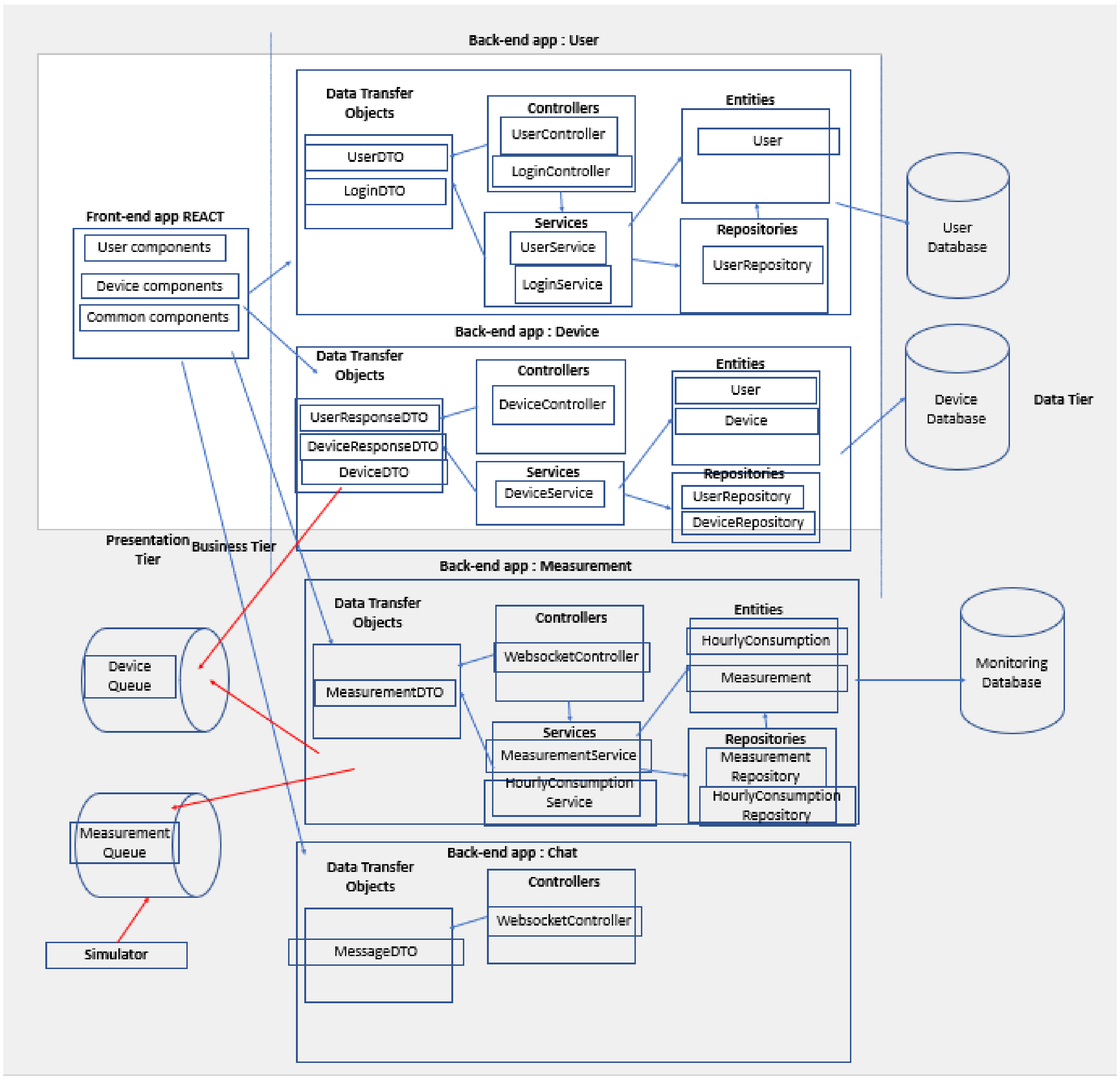
1. Objectives

Develop a chat microservice and an authorization component for the Energy Management System. The authorization component should provide secured access of users to systems' microservices. The chat microservice should allow communication between the users and the administrator of the system, allowing them to ask questions and receive answers.

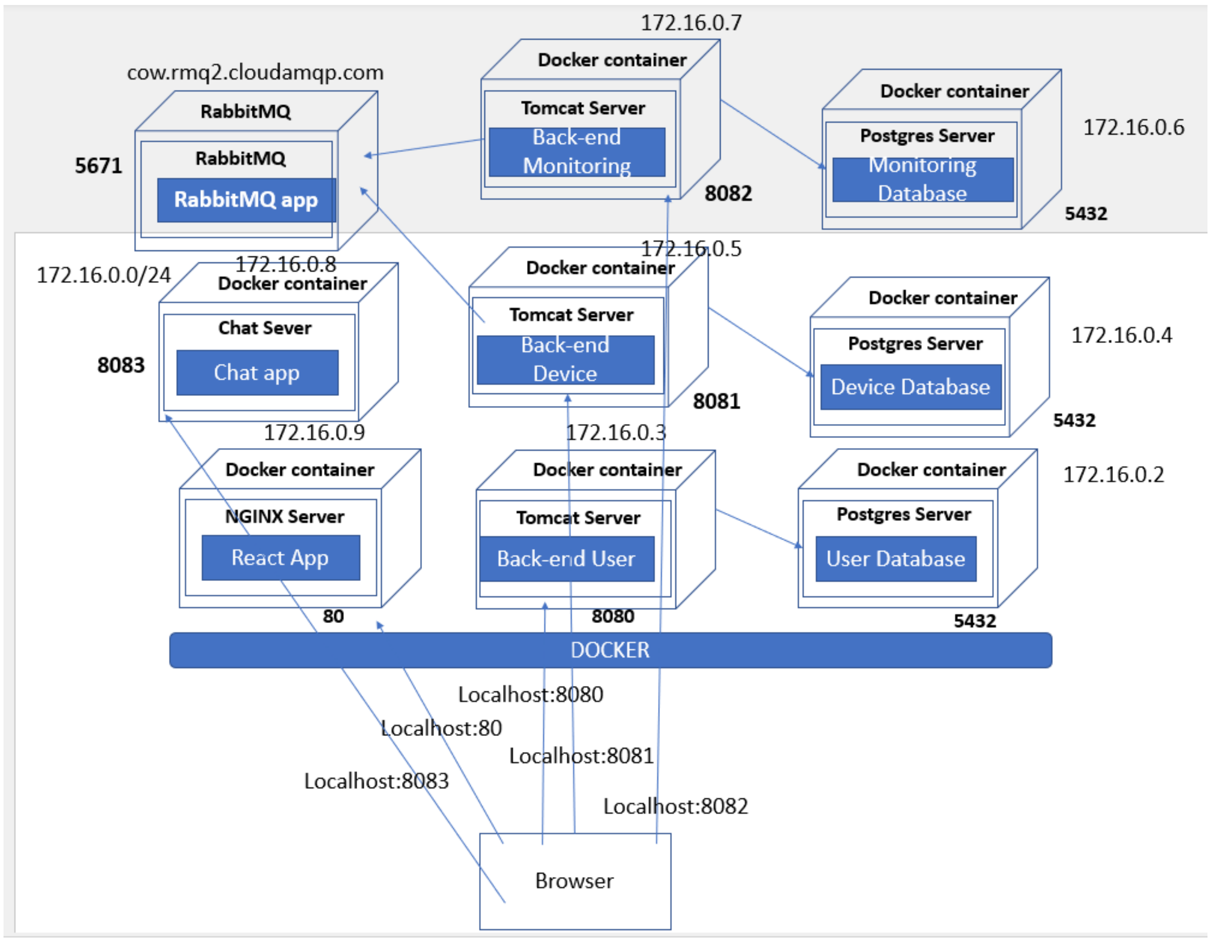
Chat microservice:
➢ The front-end application displays a chat box where users can type messages.
➢ The message is sent asynchronously to the administrator, that receives the message together with the user identifier, being able to start a chat with the user.
➢ Messages can be sent back and forth between the user and the administrator during a chat session.
➢ The administrator can chat with multiple users at once.
➢ A notification is displayed for the user when the other administrator reads the message and vice versa.
➢ A notification is displayed for the user (e.g., typing) while the administrator from the other end of communication types of its message and vice versa.
Authorization component:
➢ One of the services is chosen as authorization server (e.g. User Microservice, or a newly implemented microservice for authorization and authentication). This service generates access tokens to the client application. The tokens will be used to access other microservices.

2. Problem analysis, modelling

Conceptual architecture:



Deployment diagram:



3. Design and Implementation

In order to solve the problem of synchronizing the databases, I chose to send a POST/DELETE request from the user microservice to the device microservice any time such action is necessary. This ensures consistency between the databases, and the Transactional annotation allows for rollback in case something goes wrong. All backends are secure, device using just a simple filter that forwards to user, and user is protected by Spring Security alongside JWT token authentication.

The databases do not expose any ports outside of the docker network. This is for security purposes, as the browser does not need direct access to the databases.

4. Build and execution instructions

Build Instructions

Prerequisites

- Docker installed on your machine.

Building Images

In each folder starting with **assignment1** and **assignment2**, there is a Dockerfile responsible for building images not found by default (e.g., PostgreSQL). Use the following commands to build these images:

```
cd assignment1-user-management
docker build -t my-user-image:latest .
```

```
cd ../assignment1-device-management
docker build -t my-device-image:latest .
```

```
cd ../assignment1-frontend
docker build -t my-react-image:latest .\
```

```
cd ../ assignment2-monitoring-communication
docker build -t my-monitoring-image:latest .\
```

```
cd ../ assignment3-chat
docker build -t my-chat-image:latest .\
```

Execution Instructions

Starting the Microservices

Run the following command to start the microservices:

```
docker-compose up -d
```

This will start the PostgreSQL databases, Spring Boot applications, and the React frontend in detached mode.

Stopping the Microservices

To stop the services, run:

```
docker-compose down
```

Accessing Services

- User Management App: http://localhost:8080
- Device Management App: http://localhost:8081
- Monitoring and communication App: http://localhost:8082
- Chat App: http://localhost:8082
- React Frontend: http://localhost

Stopping the Microservices

To stop the services, run:

```
docker-compose down
```

Folder Structure

```
assignment1-device-management/
assignment1-frontend/
assignment1-user-management/
assignment2-monitoring-communication/
assignment3-chat/
```

```
.gitignore
docker-compose.yml
```

```
README.md
Doc_Pelle_Andrei Assignment_1.docx
Model_Arhitectura_Conceptuala.pptx
```

This structure organizes the microservices and relevant files.
For additional details, refer to the documentation files in the repository.