Technical University of Cluj-Napoca

Structure of Computer Systems

Device Management System

Assignment 2

TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA
ROMANIA

Teacher: prof. Cristina Pop

Teacher Assistant: Mircea Gabriel Antonesi

Student: Bogdan-Mihai Gligor

Group: 30444

Project Description

The purpose of this project was to implement a microservices-based full-stack application. The project theme is a device management system that is to be used by both administrators and regular users. An administrator can do CRUD both on **Users** and on **Devices.** They can also do mappings between the two. A regular user can see their own devices.

Solution Description

To respect the microservices architecture I needed to divide my application into 5 independent systems.

1. **Frontend**

My client application was implemented using Next.js. I chose this framework because it adds some very useful functionality on top of classic React. Like the app router, that uses the filesystem to automatically map routes to different pages. Or the capability to use React Server Components. It runs on an Alpine image, with Node.js 18.

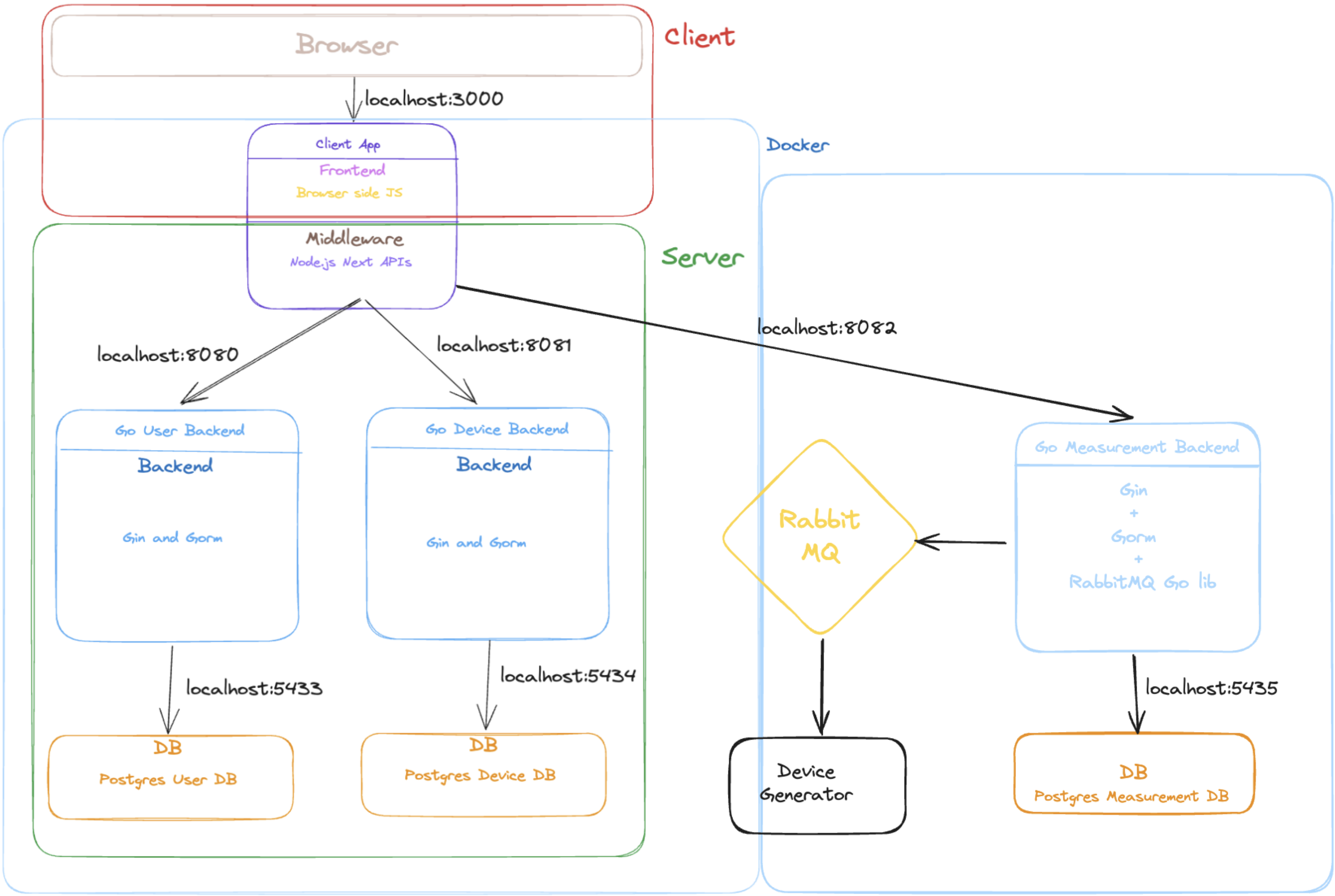
2. **Backend**

My backend application uses three Go images to run three different Go executables, one for each backend microservice. The backend services use Gin as the HTTP/HTTPS web server framework, and Gorm for database interfacing and migrations. Using these tools I created a REST API for each microservice. I chose Go because it is quite elegant to write compared to other solutions, and it has great performance. One of the microservices also uses RabbitMQ to subscribe to a device simulator.

3. **Databases**

As persistent data storage, I chose three instances of PostgreSQL databases. Postgres is an industry standard for database solutions and is also my favorite SQL database, so the choice was clear.

Deployment and Architecture Diagram


Client
Browser
localhost:3000
Docker
Client App
Frontend
Browser side JS
Middleware
Node.js Next APIs
Server
localhost:8080
localhost:8081
localhost:8082
Go User Backend
Backend
Gin and Gorm
Go Device Backend
Backend
Gin and Gorm
Go Measurement Backend
Gin
+
Gorm
+
RabbitMQ Go lib
Rabbit
MQ
localhost:5433
localhost:5434
localhost:5435
DB
Postgres User DB
DB
Postgres Device DB
Device
Generator
DB
Postgres Measurement DB