

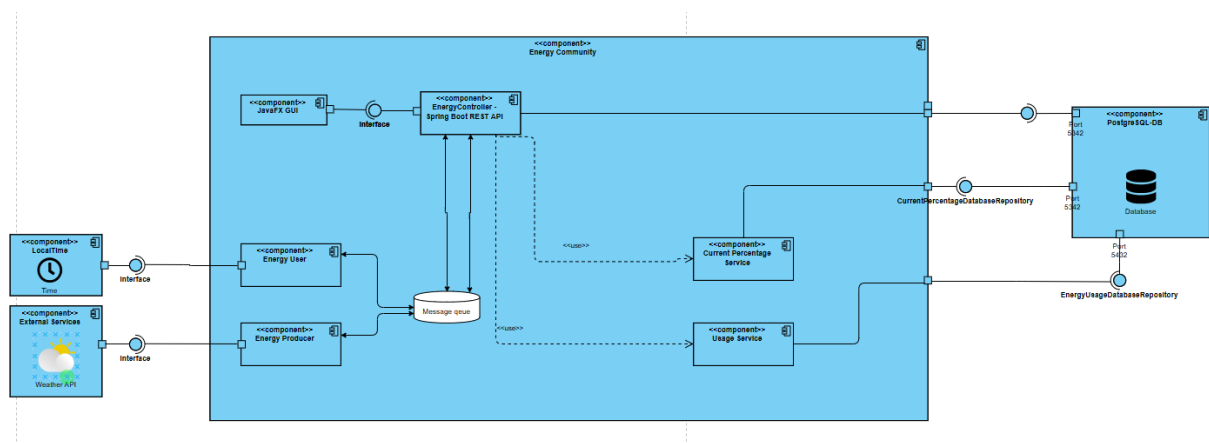
DISYS (Group-M) / ER- & Component Diagram

Inhalt

Component Diagram:	1
Project-Structure:	2
Database Configuration	2
FLYWAY-DB Migration	3
Docker und Message Broker Configuration.....	3
ER-Diagram	4
SQL.....	4

Component Diagram:

This is the component diagram of your “EnergyProject”, it illustrates the architecture of the Energy Community system. It shows how external data sources, internal services, and a Spring Boot REST API interact through a message queue and persist data in a PostgreSQL database. The JavaFX GUI serves as the main interface for users to access and visualize energy data.

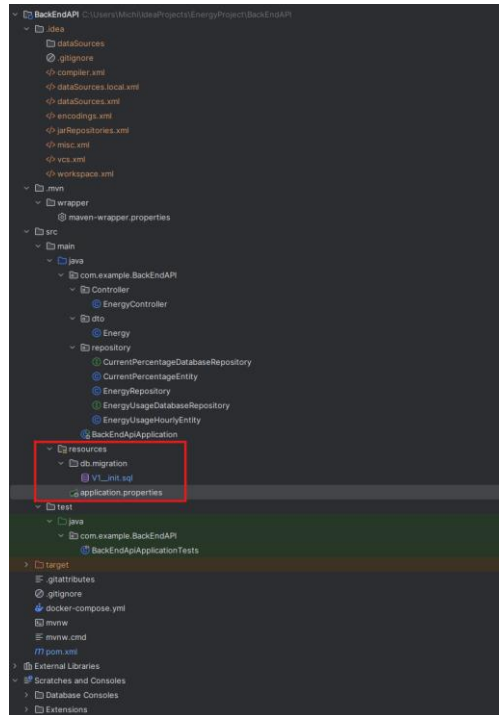


Project-Structure:

In the BackendAPI Project folder there are 2 files:

- V1_init.sql
- Application.properties

These files contain the code to establish a connection and migrate the Database.



Database Configuration

Connects to a PostgreSQL database running locally on port 5432. To establish the connection, we set the following values:

Database name: energydb

Username: disysuser

Password: disyspw

JDBC Driver: PostgreSQL driver class

```
spring.datasource.url=jdbc:postgresql://localhost:5432/energydb
```

```
spring.datasource.username=disysuser
```

```
spring.datasource.password=disyspw
```

```
spring.datasource.driver-class-name=org.postgresql.Driver
```

FLYWAY-DB Migration

This configuration enables Flyway for automatic DB migrations, by looking for migration scripts under `src/main/resources/db/migration`. As schema the public schema is used.

```
spring.flyway.enabled=true
spring.flyway.locations=classpath:db/migration
spring.jpa.properties.hibernate.default_schema=public
spring.flyway.schemas=public
```

Docker und Message Broker Configuration

A lightweight version of PostgreSQL is used as the Docker image. This is accessible via the port set on the local host (`http://localhost:5432`).

The line “5672:5672” specifies the standard port for communication between the various services. The corresponding RabbitMQ web interface can be accessed at `http://localhost:15672`.

services:

database:

image: postgres:alpine

ports:

- '5432:5432'

environment:

- POSTGRES_USER=disysuser

- POSTGRES_PASSWORD=disyspw

- POSTGRES_DB=energydb

rabbitmq:

image: rabbitmq:management-alpine

ports:

- '5672:5672'

- '15672:15672'

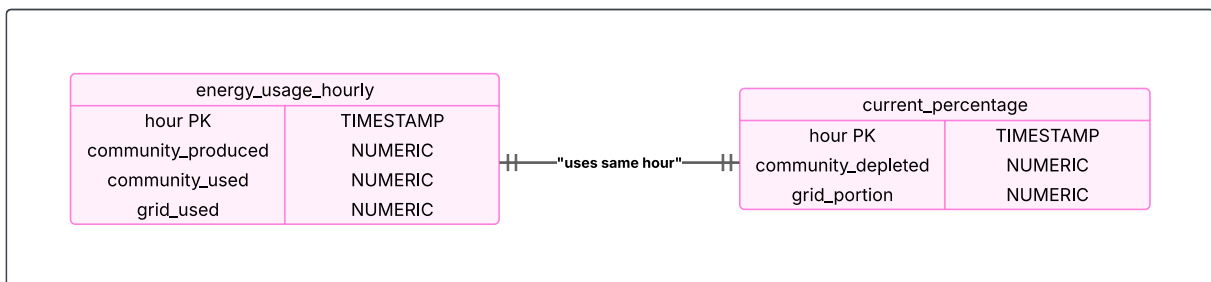
ER-Diagram

The diagram shows two relational tables for storing energy data with a 1:1 relationship via the “hour” field, which has the function of a private key.

The left-hand table stores energy usage data for users, e.g. how much energy a user has consumed and produced, i.e. contributed.

The right-hand table, on the other hand, stores the percentage of energy consumed per hour.

Energy Usage and Current Percentage Tables



SQL

```
CREATE TABLE energy_usage_hourly (  
    hour TIMESTAMP PRIMARY KEY,  
    community_produced NUMERIC,  
    community_used NUMERIC,  
    grid_used NUMERIC  
);
```

```
CREATE TABLE current_percentage (  
    hour TIMESTAMP PRIMARY KEY,  
    community_depleted NUMERIC,  
    grid_portion NUMERIC  
);
```