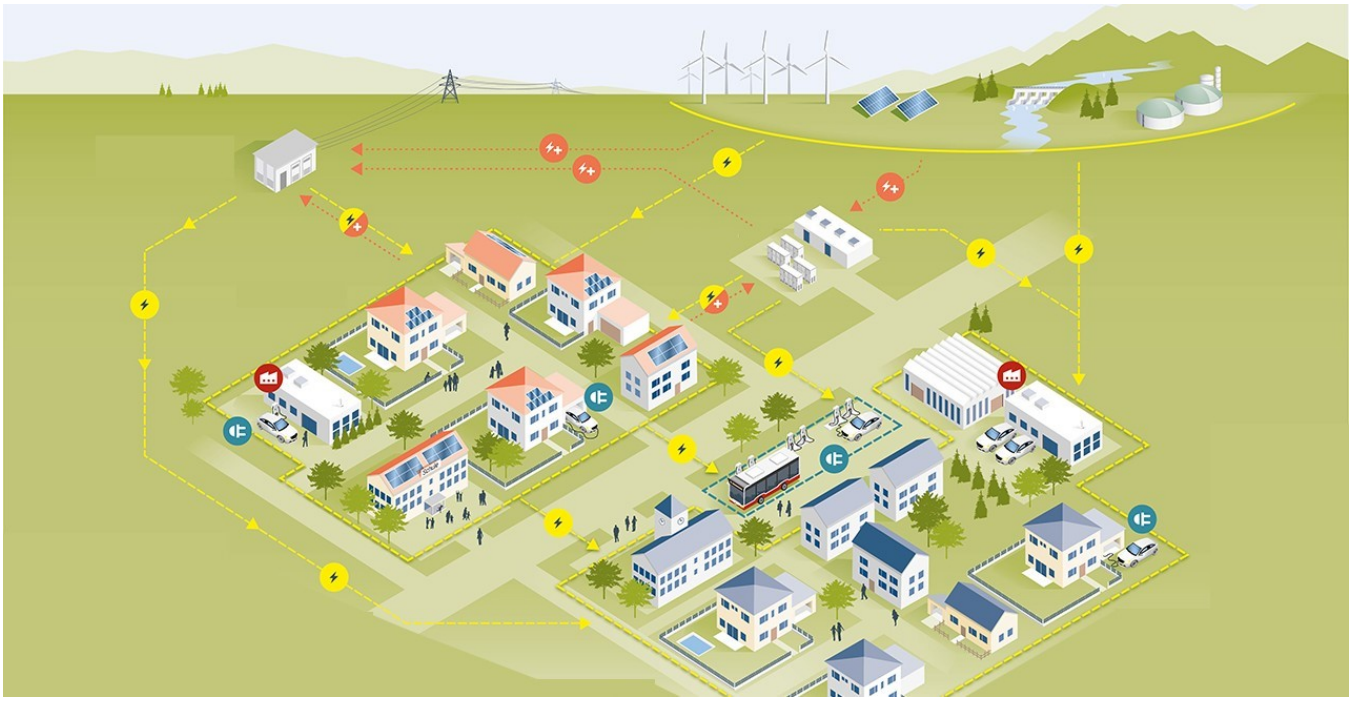


Energy communities

An energy community is an association of at least two participants for the joint production and utilization of energy. There are community producers and users and grid producers and users. You should write a system, consisting of multiple components, that shows the current energy distribution and usage.

Idea

In the center of the system is a message queue that receives energy production and usage messages. Based on these updates, a service should calculate the current community and grid usages. If a community user wants energy, the community energy pool will be used first. Otherwise the grid will deliver the energy.

After the usage is calculated, another service calculates the percentage for the current hour based on the usage.

You can monitor the current distribution of energy on a Graphical User Interface (GUI). You can also ask for historical data.

Components

You have to develop 6 components for this project. **Every component is its own application that can be started independently form the other applications.**

Community Energy Producer

A community energy producer sends the following message to the queue:

- type: PRODUCER
- association: COMMUNITY
- kwh: *the kWh produced in a minute* (e.g. 0.003)
- datetime: *the datetime of the energy production* (e.g. 2025-01-10T14:33:00)

The Energy Producer should send a message every couple of seconds with a semi random (but plausible) amount of kWh. Incorporate a Weather API to make sure more energy is produced when the sun is shining.

Community Energy User

A energy user sends the following message to the queue:

- type: USER
- association: COMMUNITY
- kwh: *the kWh used in a minute* (e.g. 0.001)
- datetime: *the datetime of the energy usage* (e.g. 2025-01-10T14:34:00)

?

The Energy User should send a message every couple of seconds with a semi random (but plausible) amount of kWh. Incorporate the time of day to make sure more energy is needed in peak hours in the morning and in the evening.

Usage Service

Every time a new PRODUCER or USER messages comes in, the database is updated. The data from individual minutes is accumulated into the corresponding hours, e.g.:

Database table before the new USER message:

hour	community_produced	community_used	grid_used
2025-01-10T14:00:00	18.05	18.02	1.056
2025-01-10T13:00:00	15.015	14.033	2.049

A new messages is processed by the queue:

- type: USER
- association: COMMUNITY
- kwh: 0.05
- datetime: 2025-01-10T14:34:00

Database table after the message:

hour	community_produced	community_used	grid_used
2025-01-10T14:00:00	18.05	18.05	1.076
2025-01-10T13:00:00	15.015	14.033	2.049

As the community user required more energy than was available in the community production pool, grid usage also increased.

Current Percentage Service

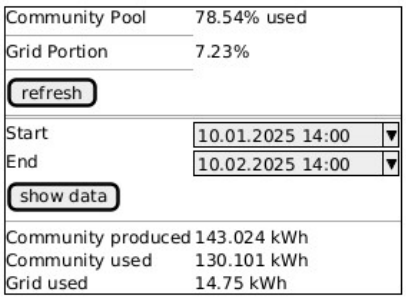
Because the usage changed, a new percentage has to be calculated.

hour	community_depleted	grid_portion
2025-01-10T14:00:00	100.00	5.63

This means the community pool is 100% depleted and the grid portion of the total energy was 5.63% . The table only hold the information of the current hour.

GUI

This information needs to be displayed somewhere. Use JavaFX to create a GUI that can display the current percentage data and historical data based on a time filter.



Important: The GUI is not directly connected to the database. The GUI uses a REST API to fetch the data.

REST API

Use Spring Boot to create a REST API with two endpoints:

- GET /energy/current
  - returns the percentage of the current hour
- GET /energy/historical?start=...&end=...

?

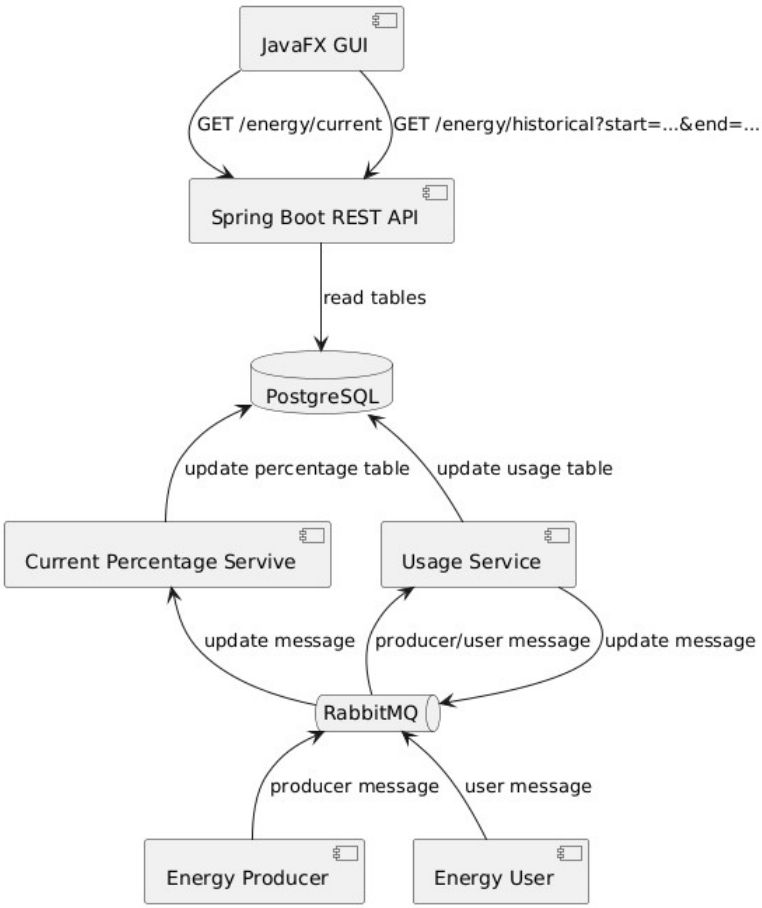
- returns the usage date for a given time period (start to end)

The Spring Boot Application is connected to the database, but can only read the information from the tables.

Example Timeline

1. Community Energy User sends minute usage data to queue
2. Usage Service picks up the minute data and updates the hour data in the database
3. Usage Service sends a message to the queue that new data is available
4. Current Percentage Service picks up the new data and saves the calculated percentage data to the database
5. GUI wants to refresh the current percentage und sends a GET request to the REST API
6. The REST API handles the request, reads the data from the database and returns the data to the GUI
7. The GUI displays the data to the user

Component Diagram



Last modified: Wednesday, 12 February 2025, 3:28 PM

?