Smart Service Development UE

Group Assignment 3 - Project Specification

Lukas D'Angelo 11713269, Patrick Eder 01607220, Benedikt Görgei 11710412

May 11, 2022

Contents

1 Scoping and Structuring

1.1 Abstract

The topic of smart home technology gained importance over the last decade, introducing the concept of networking devices and equipment in domestic areas. Increasing demand for renewable energy and efficient usage creates a need for intelligent smart-home systems to contribute to the goals of EUs *Energy Efficiency Directive* as well as creating a sustainable, reliable, scalable application framework.

On the one hand this framework integrates hardware peripherals accessed by an MQTT broker running on a Raspberry Pi. On the other hand a hardware and service engine provides logic and a database for the application. Control and state APIs are defined for standardized communication between hard- and software engines, wherein an asyncAPI is defined for data exchange between MQTT-Server and hardware engine (see Figure 1 for more details). A user can access the smart home system by entering the right credentials on a website. The web server is created with the flask python library.

A graphical dashboards provides the user with weather information requested from MET Norway Weather API v.3, status of sensors and relais and allows to set the status of actuators and define simple timer switching logic in an interactive way. Alternatively a user may also access the hardware engine directly via its exposed API. Overall this smart service is meant to be a contribution to solve the environmental and energy management challenges of the 21'st century.

2 Services and their Interfaces:

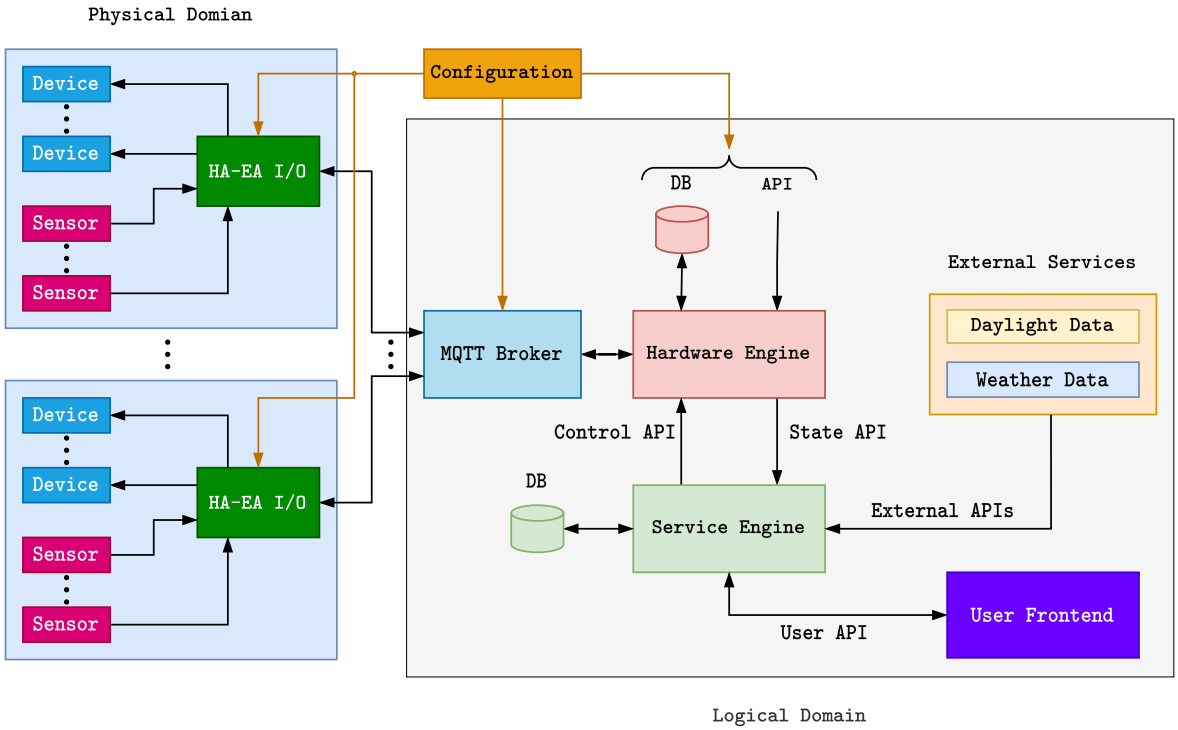

Figure 1: Concept diagram of the project

2.1 UML Component and Sequence Diagram

Figure 2 shows the main components of the project, Figure 3 illustrates a typical user interaction in form of a sequence diagram.

2.2 Physical Domain

Description: The Physical Domain is the service containing all the entities to be controlled and whose state is to be sensed, including the I/O modules, thus provide the interface between the actual hardware to sense/control and the software.

AsyncAPI (MQTT): `./physical_domain_async.yaml`

2.3 Hardware Engine

Description: Transforms the state data of the physical Domain to data usable for the Service Engine and transforms control data of the Service Engine to data interpretable by the physical domain.

AsyncAPI (MQTT): `./hardware_engine_async.yaml`

State, Control & Configuration API (REST): `./physical_domain_rest.yaml`

2.4 Service Engine

Description: Transforms state data of the Hardware Engine to visualisable data for the user frontend, transforms user commands to commands for the hardware engine, transforms data from external services to a usable format

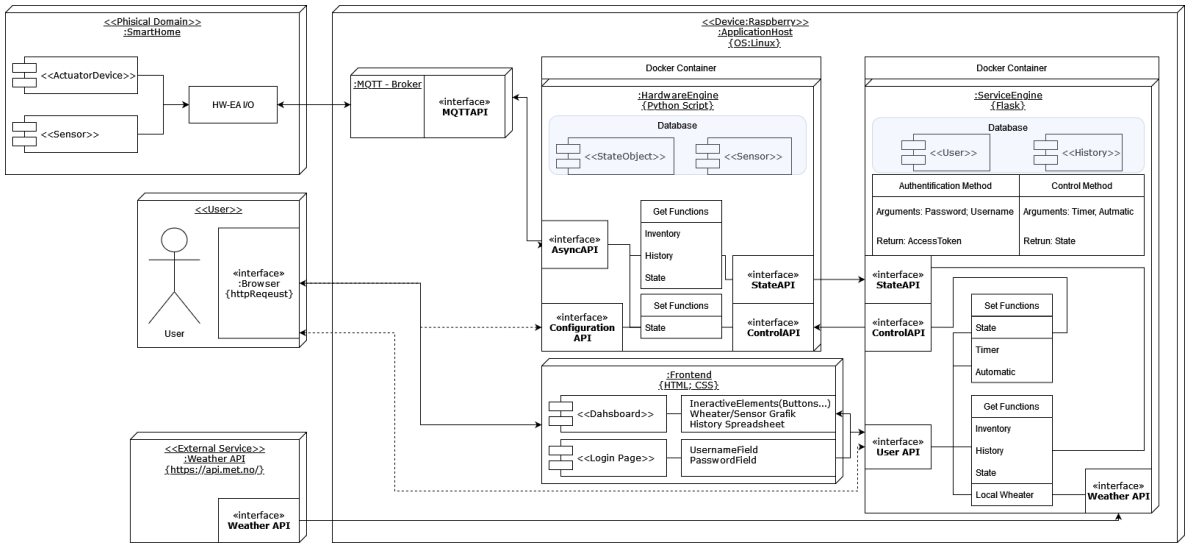

Figure 2: Component diagram of the project

2.4.1 External Services

Description: Local weather data - Get the actual weather data from the area in which the smart home service is installed. (Also displayed in the dashboard)

Wheather API (REST): `https://api.met.no/`

2.4.2 Web dashboard as Graphical User Interface

Description: Interactive elements and history in spreadsheet format is provided in a dashboard for the user as a visual interface for the user api.

User API (REST): `./User-API.yaml`

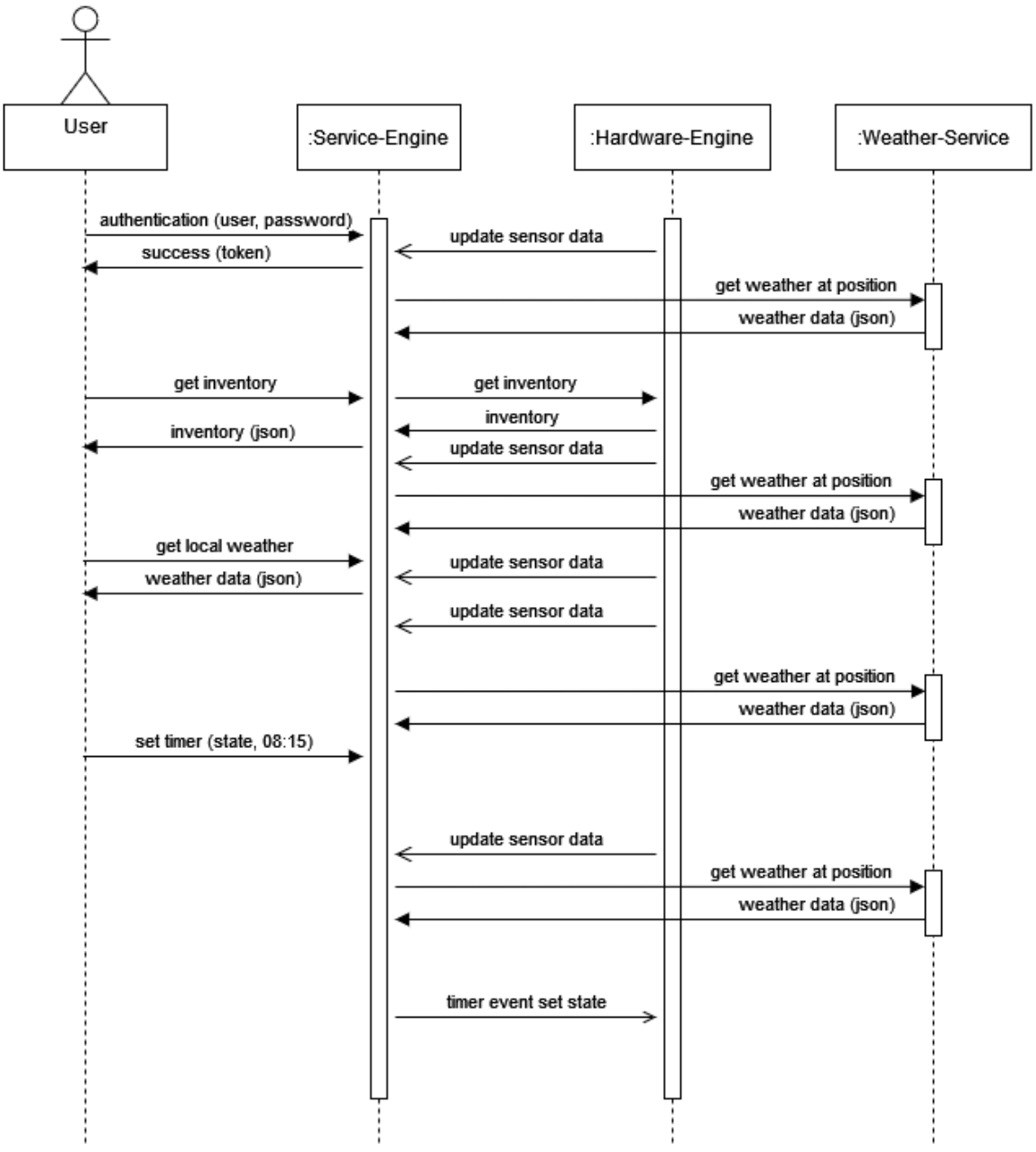

Figure 3: Sequence diagram of the project