Smart Service Development UE

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

April 26, 2022

Contents

1 Service Idea

The topic of Smart Home Technology gained importance over the last decade, introducing the concept of networking devices and equipment in the house. According to the Smart Homes Association the best definition of smart home technology is "the integration of technology and services through home networking for a better quality of living". Many tools used in computer systems can also be integrated in Smart Home Systems. In our service we want to integrate the knowledge of network programming to an existing smart home hardware. This can be seen as a smartification of the existing hardware, as we plan to develop the integration of sensors to trigger actions. To combine for example temperature sensors with a heating system, the interface needs to be defined and implemented. So the main idea is the development of an interface between hardware components and home network to the user frontend.

The system can be subdivided in three parts. First we have the hardware component, containing the smarthome services with temperature sensors and corresponding relais. Second, the backend consists of an MQTT Server, running on a Raspberry Pi and communicating with the hardware. The last element is the frontend, containing the User Interface. The overall goal is to improve the accessibility and user experience for an customer, leading to an easy to employ system.

This project can be built upon already existing hardware resources and an working MQTT server. The main motivation behind this project is the current interest in the development of smart homes, as they will gain more and more interest in the future, leading to interesting possibilities.

2 Research

The proposed service is a smart home application that includes on the one hand a web dashboard for the user of the service and on the other hand a control data collection unit for the actuators and sensors installed in a smart home environment.

Smart homes are not a new development and there are some companies in the target area of Graz where the service should be deployed (LOXON) ("Energie Graz"). Energie Graz for example already has a similar product to the proposed idea in this document. The commercial success of the service depends on its competitiveness to these already existing services. An alternative to direct competition is to work as a subcontractor for more experienced companies in the field of smart homes. Investigation on the market situation shows a potential growth for smart homes in the next few years. ("market-analysis") The increasing demand on renewable energy and the resultant extension of smart grids only further increased by reduction of Russian gas by the USA and EU could lead to new developments of integrated smart-grid/smart-home systems to meet changed energy demands and the efficient use of it ("Obligation schemes and alternative measures").

However, more research and consulting on the european and austrian market situation should be considered to find a potential market gap. In any case the construction of a demo product will be necessary to prove the ability of the service. Therefore this project should be seen as a demo service on which one can build on.

The service will depend on whether data since the activation/deactivation actuators such as heaters must be in context with the actual temperature, daytime, humidity and so on. These data can be acquired from plenty of existing weather apis ("Weather API") (Meteomatics). The hardware is already existing provided by Lukas and includes the MQTT-Server accessing peripherals running on an raspberry pi at his shared accommodation. The web-server could also run on the same raspberry or on a ("Wirtschaftshilfe für Studierende") WIST dorm server due to the fact that Patrick and Benedikt are WIST residents and Patrick has access to a working station there. An API needs to be defined for exchange of data between the web server and the MQTT server. The smart home API should also be accessible directly from the internet. A website with the login feature needs to be developed to provide the user with interactive and informative graphical elements for accessing data and control actuators. To reduce efforts python-flask boilerplate dashboards are considered. ("Flask-Dashboard")

In conclusion it can be said that the team members of this group feel up for this challenge. Parts of the system already exist and the idea is based on a well approved concept numerously used. The development of this smart home service application demo should not be much of a problem.

3 Starting to specify

3.1 The Physical Domain

As already mentioned, some of the key building blocks of our service are preexisting HA-EA I/O modules with open and pre-defined APIs. These modules can be used to control lighting, heating plants, irrigation plants, they can be used to sense the state of switches and motion detectors and can be used to gather environmental data such as room temperature and so on.

Definition: The `Physical Domain` is the set containing all the entities to be controlled and whose state is to be sensed, including the I/O modules.

3.2 The Logical Domain

In order to interface with an arbitrary number of entities and thus, an arbitrary number of I/O modules, creating an abstraction layer of the `Physical Domain` seems reasonable. We call this abstraction layer (service) `Hardware Engine`, it has the following tasks:

- Device and I/O inventory
- Interfacing with the I/O modules over MQTT
- Providing a configuration storage and state storage of the `Physical Domain`

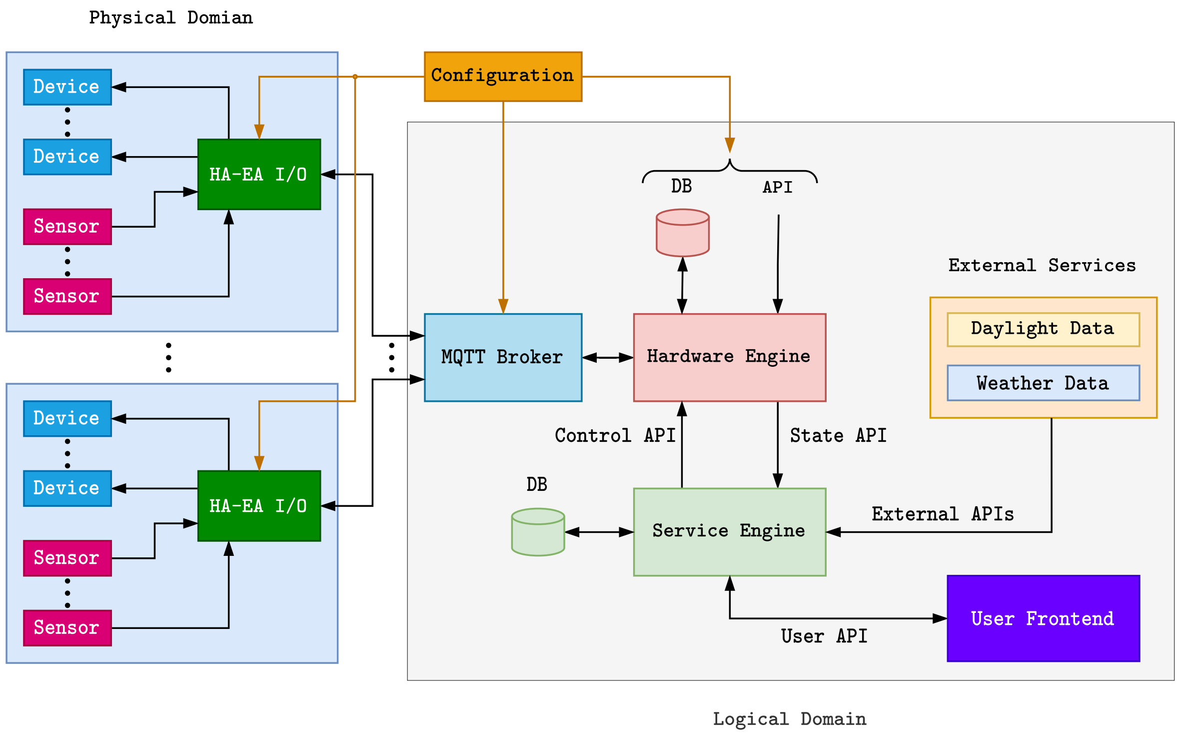

Figure 1: Concept diagram of the project

- Providing a configuration API (REST)
- Providing a control API for controlling entities (REST)
- Providing a state API for retrieving the state of entities and the inventory (REST/socket)

The `Hardware Engine` on its own does not provide any automation functionality nor an appropriate interface for human entities. These aspects are handled by the `Service Engine`, more specifically, it fulfills the following tasks:

- Interfacing with the `Hardware Engine`
- Providing a data store for entity state storage and history
- Providing a data store for its own parametrisation
- Providing a `User Frontend`
- Interfacing with `External Services` for auxiliary data
- Providing a rule engine for automation

Definition: The `Logical Domain` is the set of services which control and sense the state of the physical entities while interfacing with the `Physical Domain` and the User.

3.3 The Configuration

Information is required to describe the `Physical Domain` (e.g. the amount and type of I/O) and to describe the interface between the logical and physical domain (e.g. 802.11 connection parameters, MQTT credentials). A subset of this information is used and stored in blocks where it is required.

3.4 Data Sources, Sinks & Transformers

Figure 1 shows the context of the data sources, sinks and transformers.

Table 1: Primary data sources

Data Source	Domain	Description
Digital Inputs	physical	Digital Inputs on I/O modules
Temperature Sensors	physical	Digital Temperature Sensors on sensor BUS
Daylight Data	logical	Information about sunrise & sunset for automation
Weather Data	logical	Information about local weather for automation
User Interface	logical	User commands and user defined automation rules

Table 2: Primary data sinks

Data Sink	Domain	Description
Digital Outputs	physical	Digital Outputs (relays) on I/O modules
User Interface	logical	Entity state & history, device inventory, user commands

Table 3: Data transformers

Data Transformer	Domain	Description
Hardware Engine	logical	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
Service Engine	logical	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.

3.5 Potential Information Flow

In Figure 1 the arrows connecting the various blocks illustrate the potential information flow in our system. Let us consider the following example to sketch the information flow in practice:

- We want to control the temperature of a room
- An electrical valve is used to control the warm water flow of the readiators
- A temperature sensor is used for temperature feedback
- The sensor and the valve are connected to the I/O module

The user sets the following conditions for controlling the temperature of the target room in the user interface:

- A target temperature (e.g. 21°C)
- A time interval when heating should be done (e.g 08:00 - 20:00)
- The weather conditions (e.g. not sunny, outdoor temperature < 5°C)

This user input is transformed by the Service Engine into an automation rule. The Service Engine collects all the information required to decide if the conditions of the user are met and the heating valve should be activated:

- It queries the room temperature value over the state API of the Hardware Engine

- It queries the outdoor temperature and wheather over an external service
- It verifies if the current time is inside the heating time interval

If the conditions are met, the Service Engine makes an API call to the hardware engine which then ensures that the I/O module activates the target valve.

The Service Engine updates the values for the conditions periodically such that the heating is controlled as the user specified.

3.6 References

"Energie Graz." Smart Home, Accessed: 04.05.2022
https://www.energie-graz.at/egg/energie-dienstleistungen/energie-sparen/smart-home.

"Flask-Dashboard." Datta Able, Accessed: 04.05.2022
https://appseed.us/admin-dashboards/flask-datta-able.

LOXON. "LOXON Home Automation." elektro-gschanes, Accessed: 04.05.2022
https://elektro-gschanes.at/.

"market-analysis." European Smart Home, Accessed: 04.05.2022
https://www.idc.com/getdoc.jsp?containerId=prEUR147603321.

Meteomatics. "Meteomatics." Free Basic API, Accessed: 04.05.2022
https://www.meteomatics.com/de/entry-level-api/.

"Obligation schemes and alternative measures." European Commission, Accessed: 04.05.2022
https://energy.ec.europa.eu/topics/energy-efficiency/energy-efficiency-targets-directive-and-rules/obligation-schemes-and-alternative-measures_en.

"Weather API." Weather API, Accessed: 04.05.2022
https://www.weatherapi.com/api-explorer.aspx.