ON THE ROAD TOWARDS A SMART URBAN DISTRICT SUPPORTED BY A MOCK PLATFORM

Frank Elberzhager, Patrick Mennig and Phil Stüpfert
Fraunhofer Institute for Experimental Software Engineering IESE
Fraunhofer Platz 1, Kaiserslautern, Germany

ABSTRACT
The world is becoming more and more connected, which is also true for smart cities. The digital transformation is making rapid progress, and part of this are digital ecosystems, consisting of a platform and digital services that support several stakeholders with digital services. In the context of smart cities, we are currently developing a climate-neutral smart urban district and use digital solutions to address topics such as mobility, energy, smart home, and communication among the citizens involved. However, new ideas for services are challenging to invent, develop, and test in our situation, as this particular urban district is currently under construction. This is why we have developed a mock platform, which makes it much easier to evaluate new ideas fast, as qualities of a productive environment such as security or performance are not as highly relevant as an easy-to-use environment. We present details of our mock platform and our experiences in applying it to date. Such a solution is also beneficial for other practitioners and researchers in other smart cities and smart urban districts who are developing a digital ecosystem with new digital services.

KEYWORDS
Digital Ecosystem, Smart Urban District, Platform, Mock Platform, Digital Services

1. INTRODUCTION
Nowadays, we live in a highly connected world. Connected cars, homes, and factories, smart and mobile devices, and even combinations of these are only some examples. In this context, topics such as the Internet of Things (IoT), digital ecosystems, or cyber-physical systems have become established. With respect to smart cities, many of these topics are highly relevant. Doku and Rawat provide an overview of smart city applications (e.g., smart transportation, smart energy, smart infrastructure), and refine them into many more smart services [Doku19]. Wang et al. [Wang21] provide a literature survey in the area of energy management solutions in the IoT showing that many topics are currently under research, and some are already being evaluated. With respect to our own research agenda with a focus on digital ecosystems, several open issues exist, such as: How to design, develop, deploy, and operate a digital ecosystem? What concrete technology to use? Which stakeholders to involve? How to build a platform that other parties can use?

On the way towards smart cities, one intermediate step is to address such topics in a smaller part, such as a local urban district. We are currently part of a big research project that strives towards realizing a climate-neutral urban district. In this project, several different players are involved, such as the local city government, research partners, and companies. A former industrial complex about 18 ha in size is being revitalized and rebuilt for this. Many sub-topics are currently being worked on, such as energy, mobility, smart home, or social-technical topics. We are responsible for creating a digital ecosystem for this urban district that supports several different stakeholders in behaving in a more climate-friendly manner. One focus is on the citizens who will live and work in this urban district, and who we want to support in the first phase. Examples of digital solutions in this digital ecosystem are a game to make citizens aware of existing climate-neutral mobile services, an app to enhance communication about climate-relevant topics in such a district, and advice on how to save energy with a digital pet. A major challenge with respect to our urban district is the fact that it is currently under construction. This means that only a very limited number of stakeholders are already known, and we have to find ways to (1) anticipate and gather requirements of future citizens and (2) further stakeholders that will use our digital ecosystem. Furthermore, we need to find ways to
deploy, apply, and evaluate our digital solutions. We have performed different kinds of project-internal and external workshops, done a literature research, and used other techniques to gather such requirements. In this publication, we focus on the second aspect, i.e., we provide one solution for coming up with and testing digital services with concrete users.

The central part of our digital ecosystem is a platform with several basic services (such as a central login or user administration) and higher-level services such as those mentioned above. As this is a productive environment, qualities such as performance or security are of high relevance. Consequently, running services need a certain level of quality in order to be allowed to run, which also results in some deployment effort. As, among other things, these conditions are obstacles in terms of having an efficient environment for a fast evaluation of new ideas and early versions of new digital services, we had to find a different way. Therefore, we developed a so-called mock platform. This is a simpler version of the real urban district platform that does not care about, for example, privacy issues, but is similar enough to the real platform to give us a realistic evaluation environment.

In this publication, we provide some details on our current mock platform and its technical details (Section 2). Then we show examples of how we are already using the mock platform and share some initial experiences (Section 3). Finally, we conclude our paper and give an outlook on future ideas (Section 4).

2. MOCK PLATFORM

2.1 General Idea and Benefit

Our main goal is to create a way to connect different (also external) applications through one platform. However, this should be as simple as possible when developing new applications, which is why we have specifically neglected requirements such as privacy, security or safety. For this reason, we also chose a very simple and lean communication protocol. In order to address the above-mentioned problems with respect to our need for fast evaluation of new ideas for digital services in our smart urban district ecosystem, we developed a mock platform. On the one hand, this mock simulates a real platform with basic functionality, on the other hand, makes it much easier for us to deploy and test new digital services on the other hand.

One major challenge when connecting an application to the platform is the correct semantic and syntactic communication. For this, we decided to us asynchronous communication, which also enables us to facilitate connections between different digital services. We use the publish-subscribe pattern, which makes it possible to publish messages easily via an interface without the sender having to know the exact recipient or their exact address. Basically, any application can receive all messages. In order for a potential recipient to receive the correct messages, the sender publishes a message on a specific topic. A recipient can now subscribe to this topic and will then receive only the messages published on this topic.

2.2 Insights from Architecture and Technology

The core of the mock platform is an MQTT broker, serves as a central asynchronous communication interface according to the publish-subscribe-pattern (see Figure 1). The MQTT protocol is very common, especially in the IoT domain. In addition, we offer various basic functionalities in the form of services, that are required again and again, especially in the platform context but also generally in prototype creation. For example, on our mock platform, a user service offers the possibility to create users who receive a unique global ID in the ecosystem. In addition, there is the storage service, which serves as a central storage location and enables storage of configurations of applications, which can then be loaded when an application is started. There are a number of other services like calendar service that are intended to shorten the development time.

The messages via the MQTT broker always have an event character. This means that messages can be sent by a sender whenever something has happened in its own application. For example, every time a user registers, the user-service publishes this information on the MQTT broker.
To make messages not only easy to receive technically, but easier to process in addition, we have introduced so-called shared-topics. These summarize some groups of data containing a similar information. It is defined precisely on which topic this information must be sent and what form the message must have. For example, we have defined the smart home shared topic for the smart home domain. This includes messages containing information about sensor values, such as a room’s current temperature or the opening status of a window. It is exactly prescribed how the topic is structured, so that one can read from it which room's data it is about. The message itself must also follow a precise schema which is defined for every shared topic. For example, the message must be formatted as a JSON string (see Figure 1, left part).

![Figure 1. Example communication via MQTT and high-level architecture of the mock platform and communication](image)

With this precise specification, a sender always knows how to publish information from a shared topic. In addition, a receiver who has previously subscribed to a shared topic can always be sure that they can process this message. However, the communication path via the MQTT broker should not be used in all cases. For synchronous requests, where it is important for the sender to know who the receiver is and whether the receiver has received and correctly processed the information, this path is unsuitable, since the MQTT protocol does not forward receipt confirmations from the receivers back to the initial source. For example, it may be important to an application that an object has been successfully stored on the storage-service. Since this is not possible with the asynchronous communication of the MQTT broker, we use a synchronous communication in these cases, which takes place directly between application and service. This is realized with a Rest-API, which is provided by a service.

To enable a developer to connect an application to the mock platform as quickly as possible, we provide software development kits (SDKs) in common programming languages (Java, Java-Script, Type-Script, Python), which provide easy-to-use methods that handle the correct communication with the MQTT broker. Furthermore, we provide event builder methods for all shared topics, which generate valid events including the correct topics. All this makes it possible to realize communication without much effort. For the MQTT broker we use Amazon-MQ, which provides a native MQTT interface. The services are deployed on servers in Docker containers for cost reasons, so it is easy to exchange and update the services.

3. USING THE MOCK PLATFORM – INITIAL EXPERIENCES

Within our organization, we have used the mock platform to develop prototypes of interconnected applications. It has allowed us to test the vision of such a networked smart city application within a short period of time by reducing the development efforts, mostly for the user interface, and still providing us with a means to test the interconnectedness of applications. Using pre-defined shared topics is simple due to the event broker architecture, which in turn helps to realize distributed but interconnected applications.
One example of such an application is the *smart home simulator*. Part of our ongoing research efforts for the climate-neutral smart urban district is concerned with the potential use of smart home technology, i.e., sensors and actuators that help to reduce the energy consumption of residential homes and business premises. The *smart home simulator* is a web-based 3D visualization of an apartment with several smart home sensors and actuators, such as controllable light bulbs, or window sensors that register open and close events. Even though the general concept of such smart home appliances could be demonstrated in any building, the *smart home simulator* and the mock platform can be used to simulate the actual “smart homes” as they are being planned in this concrete smart urban district to help investors and future residents understand the impact of this technology. Physical smart home sensors and actuators can be connected to the mock platform and can publish their state through shared topics. The *smart home simulator* visualizes these states in its 3D user interface. Users can interact with the simulated smart home and trigger actuators. Through the connection with the mock platform, other applications can respond to status changes by listening to shared topics, and novel smart home solutions can be prototyped rapidly, either with or without actually using physical smart home appliances.

The update events that can be issued through the *smart home simulator* are used within another prototype application that harnesses the simple interconnectivity through the mock platform’s event-based architecture. The *environmental hints* application provides users with feedback on the environmental impact of and possible alternatives for their daily life activities (e.g., the effects of heating their homes above a certain temperature on energy consumption and related costs). The better these hints fit the actual activities of a given user, the better the perceived value of such hints, leading to a higher chance that the user will change climate-relevant habits. In order to match the correct hints with the users who actually performed certain activities, the activities need to be registered by the digital smart city platform. The mock platform gives the ability to prototype such activities through fake events issued by the *smart home simulator*, a prototype calendar application, and *public transport information*, and to assign them to user personas in the *user service*. The hints are generated in the *environmental hints-giver service*, which is connected to the mock platform, analyzes various shared topics for relevant user actions, and assigns hints to specific user accounts based on the events that other applications emit for that particular user account. Test users interact with the *smart home simulator*, e.g., by increasing the temperature in the living room and then indicating they are leaving the house, which results in a certain hint given in the *environmental hints* application. This gives us real feedback on the perceived value of such hints.

### 4. CONCLUSION AND OUTLOOK

We are developing a digital ecosystem for a smart urban district in order to support climate-neutral behavior. As this urban district is currently under construction, we face the challenge of how to develop digital solutions for the future citizens and other stakeholders working and living in that area. The future digital ecosystems will basically consist of a platform and several basic and enhanced digital services such as smart home or mobility services. However, in order to be able to develop new ideas and deploy them as services in the environment, we decided to first develop a so-called mock platform, as this suits our needs much better, especially with respect to speed (i.e., fast deployment and evaluation). The mock platform is already being used by project members, but has also been applied in events such as hackathons where the intention is the same: try out new ideas fast. The mock platform simulates our smart urban district environment and reflects the future area well.

In the future, we will develop this mock platform further, e.g., with more support during the deployment and evaluation phases, and increase its usability. In addition, more digital service prototypes should be developed and evaluated to prepare for the real digital smart urban district.

### ACKNOWLEDGEMENT

The research described in this paper was performed in the EnStadt:Pfaff project (grant no. 03SBE112D and 03SBE112G) of the German Federal Ministry for Economic Affairs and Energy (BMWi) and the Federal Ministry of Education and Research (BMBF). We thank Sonnhild Namingha for proofreading.
REFERENCES

Doku, R., Rawat, B. R. 2019, Chapter 8 - Big Data in Cybersecurity for Smart City Applications, Editor(s): Danda B. Rawat, Kayhan Zrar Ghafoor, Smart Cities Cybersecurity and Privacy, Elsevier, Pages 103-112