Mobile application for energy management in smart buildings

Lazar Berbakov, Marko Batić, and Nikola Tomašević
University of Belgrade, Institute Mihajlo Pupin, Volgina 15, Belgrade, Serbia
lazar.berbakov@pupin.rs, marko.batic@pupin.rs, nikola.tomasevic@pupin.rs

***Abstract*—This paper discusses Smart Energy Manager (SEM) mobile application for energy management in smart buildings which is developed as a result of the ongoing H2020 projects InBetween and Sofia. SEM app belongs to the visualization layer of the InBetween platform serving as an interface from the system towards the end users with the aim to help them to identify their energy wastes, teach them how to conserve energy and steer their behavior to be more energy efficient. The application is based on Android OS, and it relies on a backend built with CakePHP framework which is used to integrate the application with the rest of the InBetween platform, its main data storage and advanced energy analytic services. Finally, we present the main functionalities of the app along with the data scheme which is used to store users and deployed equipment related data.**

I. INTRODUCTION

Advances in microelectronics, sensing technology and wireless communications have enabled mass production of low-cost, miniature yet computationally capable devices. Such devices can be configured to form the so-called Internet of Things (IoT) empowering objects in our environment with communication, sensing, and actuating capabilities thus enabling their usage in a myriad of innovative applications.

It is expected that the steady growth of human population and technological advancements will require either to increase the energy production or to force general community to be more energy efficient and less wasteful. Residential sector has been identified as one of the most energy demanding, providing therefore an opportunity for huge energy savings. In order to be able to fulfill these goals in reality, it is necessary to deploy different devices capable of monitoring energy consumption and user's behavior in the households. The measurements obtained by such devices can be collected by means of ICT infrastructure and after suitable analysis, the results can be presented to the end users instructing them how to act to save energy and reduce the overall cost, without interfering much with their comfort and daily routine.

This opportunity has already been recognized by different projects [1][2] and companies which are offering innovative hardware accompanied by intuitive mobile and web apps. Energic mobile app [3] helps user in saving energy by selecting the most pertinent data and provide an overview of situation by adapting the presented information to specificities of each particular user profile. Smappee [4] and eGreen [5] on the other hand are provided via web-based interface and allow user to gain insights and control over energy consumption and production by accessing real-time information on electricity, solar, gas and water.

An extensive amount of research has been focused on smart home and building automation interoperability, as elaborated by Jarvinen and Vuorimaa [6] and reported by Jiang et al. [7]. There are many possible typologies of architectures to improve interoperability with smart home devices as investigated by Capitanelli et al. [8].

The mobile application which will be presented in this document is a result of the ongoing H2020 project InBetween which aims to help users to identify their energy wastes, learn how to conserve energy, motivate them to act and finally help them to actually carry out energy efficient practices.

II. RESEARCH QUESTIONS

In this paper, our goal is to present the architecture of the InBetween project platform with the particular focus on the mobile application – SEM (Smart Energy Manager) which will be used as the main interface towards the end users. The majority of mobile apps only show the measured data to the users without actually explaining the meaning of the data them, which can result in poor app retention rate. The aim of SEM app is to provide users not only with the measurements obtained by the monitoring devices deployed in their households, but also to provide them with actionable suggestions on how to improve their energy consumption and reduce the wastes. By taking such an approach, the app retention rate is expected to be improved since the application suggestions will be easy to understand and the app can even be used by users which are not able to interpret the raw measured data themselves.

III. METHODOLOGY

InBetween project envisions two main development phases. Firstly, during the so-called baseline period after the monitoring devices have been deployed, the InBetween platform will only collect the consumption and user behaviour data without providing any suggestions to the users, in order not to interfere with their normal energy usage habits. The baseline period will last for one year, and its aim it to serve as a benchmark for the subsequent year where the InBetween platform will be fully operable, with all the energy analytic services running.

During that second phase, the InBetween platform will use SEM mobile app as a communication means towards end users to send them useful advices based on measured data relevant to the specific user. The advices on energy management will be given in a nondisruptive manner without compromising user's comfort and daily activities, at the same time taking advantage of capabilities of the platform to infer user behaviour and its energy related habits. In order to achieve this goal, a user will be compared to its neighbours in a fair manner taking advantage of the positive social pressure [9] in order to achieve the behaviour change. Different KPIs such as energy performance will be used, to enable fair benchmarking among users instead of simply using just energy consumption. Energy performance will be normalized against different objective parameters such as size of household, number of occupants, construction material properties, etc. In such a way, by comparing the energy consumption, cost and other relevant KPIs during these two periods, it will be possible to measure the impact of the InBetween platform on the energy savings. Nevertheless, in order to ensure early user engagement, even during the baseline period, the users will be allowed to view some of the measured data via SEM mobile application.

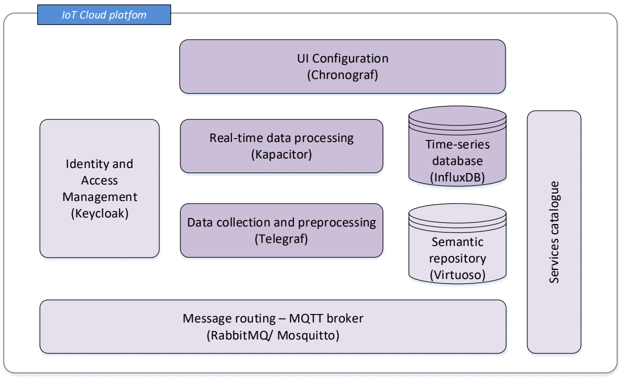

Figure 1. InBetween IoT Cloud platform

IV. Solution/Discussion

A. InBetween cloud platform

In this section we provide a description of InBetween cloud platform which forms the basis of the SEM mobile app. The main components of the cloud-based integration layer (see Figure 1) aim to integrate the advanced energy services and visualisation with the corresponding monitoring and control devices (sensors, smart meters, occupancy detectors, smart relays, etc). It ensures the provisioning and configuration of sensors, smart meters and actuators to the overall system and enable collection, exchange and processing of data. In Figure 1, we highlight the InfluxData TICK Stack, composed of Telegraf, InfluxDB, and Kapacitor. These components can be configured and some tests can be performed in order to verify that sensor successfully acquire data by their visualisation via a test GUI (Chronograf). InfluxData TICK stack is an open source platform for managing IoT time-series data at scale, that makes it particularly suitable for InBetween platform development. It allows users to manage metrics, events, and other time-based data.

1) *Telegraf*

Telegraf is a metric collection service which can be run independently from other components of TICK stack. The main purpose of Telegraf is to collect metrics from different inputs and write them into a number of outputs. It is a plugin-based component which can be easily configured and extended with custom input/output interfaces. In addition, data collected by Telegraf can be sent in different formats such as: JSON, Value, Nagios, InfluxDB Line protocol, and many others. In InBetween, Telegraf is used to collect the data sent by field level devices (sensors and smart meters) to MQTT broker and also to transform and save such data in Influx database.

Figure 2. Chronograf dashboard

2) *InfluxDB*

Time-series databases such as InfluxDB are known to deal with high workloads and requirements. In particular, they need to be able to collect a large amount of data points per second. Besides, InfluxDB needs to perform real-time queries across large data sets in a non-blocking manner, and to down-sample and evict high-precision low-value data. Because of all of these benefits, InfluxDB has been chosen as a main data repository for data collected by monitoring devices deployed in the users' home, such as: energy consumption, temperature, occupancy, etc.

InfluxDB uses a SQL-like language for interacting with data – InfluxQL, which allows large amount of data to be retrieved and analysed in real-time. InfluxDB schema can be described as follows:

- Database – represents a logical container for different time-series data, users, retention policies, and continuous queries.
- Series – represent the collection of data that share common measurement, tag set and retention policy.
- Measurement – stands for the data stored in the associated fields.
- Tag keys and values – used to store metadata. Tag keys are indexed so that queries that are performed on tag keys are faster.
- Field keys and values – used to store metadata and the actual measurements and can be of different types: integer, float, string and boolean. Fields are not indexed and each field value is always associated with a timestamp. Queries on field values scan all points that match the specified time range and, as a result, are not performant.
- Retention policy – describes for how long the data will be kept in the database. By default, it is set be infinite (no data are removed)
- Continuous query – an InfluxQL query that runs automatically and periodically within a database. It can be used to aggregate older data in order to save persistent storage.

3) *Chronograf*

Chronograf is a component of TICK stack which can be used along with other components of the TICK stack to manage databases, users, visualise collected data and create alerting and automation rules. It supports multi user configuration with access level control. An example of Chronograf dashboard showing data collected from one of the pilot sites is given in Figure 2 .

4) Kapacitor

Finally, Kapacitor is the last component in TICK stack whose main goal is to support real-time data processing and creating alerts when anomalies in data are detected. It can be configured to process both data streams in real-time or the recorded data in batch fashion. In addition, it can provide any embedded transformation which is offered by InfluxQL query language. Kapacitor can use scripts with lambda expressions in order to configure data transformations and to define logic conditions which can be used to filter data

5) MQTT Message broker

MQTT is a lightweight message transfer protocol based on publish/subscribe messaging pattern. It effectively decouples the sending party (publisher) from the receiving one (subscriber) bringing many benefits:

- One Publisher could send messages to many different Subscribers
- One Subscriber could receive messages from many different Publishers
- It is not necessary that Publishers and Subscribers are aware of each other.
- Implementation of the publisher and subscriber parties independently from each other.

The aforementioned properties of MQTT, make it ideal for distributed systems consisting of a large number of constrained devices, such as Internet of Things applications where small code footprint and limited network bandwidth are usually limiting factors for implementation. In publish/subscribe pattern, publishers and subscribers never communicate directly to each other. Instead, a so-called message broker handles connection between them. The main function of the message broker is to filter all the messages received by different publishers and distribute them only to the interested subscribers. As a result, publishers and subscribers are decoupled and do not have to be connected to the message broker at the same time.

MQTT uses message filtering based on a message subject. Each published message is assigned to a topic, which is then used by the broker to send the message only to the clients which have been previously subscribed to that topic. Topic is a string which consists of one or more topic levels separated by a forward slash (e.g. building1/floor2/bedroom/temperature). This allows the receiving party to subscribe only to a part of the topic, and to receive the messages published to topics below that level (e.g. subscribe to topic building1/# to get all the messages originating from that specific building).

B. SEM mobile application

SEM app (shown in in) is based on Android OS and serves as the main platform interface towards the end user, allowing it to observe measured data, obtain suggestions and even perform some of the control actions automatically (e.g. switching the appliances remotely). After the user performs login by using username and password combination, it is presented with the main app dashboard (see Figure 3).

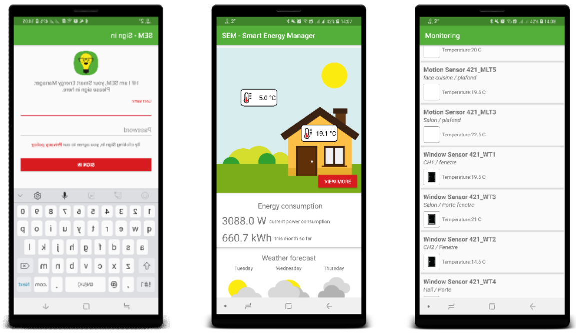

Figure 3. SEM Mobile application user interface

In the upper part of the screen, the user is given information about average temperature (across all temperature sensors deployed in household) and the current outdoor weather conditions (obtained by external weather service weatherbit.io). Next, the user is shown the information about current aggregated power consumption (obtained from the smart meter), and the total energy consumption for the current month. Finally, weather forecast for the next three days is shown. Although SEM is primarily aimed at energy management, we decided to include additional functionality in the form of weather forecast, in order to engage users early. This is done especially since during baseline period we are not able to show them much of the collected data, in order not to ruin the benchmark comparison.

SEM app is not self-sufficient, and it requires a backend component which will serve for authentication and authorization of the users, transformation and aggregation of data obtained by monitoring devices, and those coming as a result of the advanced energy services. The backend is based on CakePHP, an open source web development framework following MVC pattern, allowing easy customization and extensibility of most aspects of the resulting software. The user related data (login, deployed devices, access rights) are stored in a MySQL database with the schema given in Figure 4, whereas collected measurements are stored in Influx database which is particularly suitable for time-series type of data.

As can be seen in Figure 4, the users are grouped into groups in order to enable group-based access control. Currently, we envision three user groups:

- Admin – mobile app administration, creation of user accounts, assigning of gateways to specific users, etc.
- Household admin – household user with permission of monitoring and control actions
- Household user – household user with monitoring only permission

Each user is assigned to one Device Group, an entity which can group one or more gateways and the corresponding devices (smart meter, smart plug, window sensors, temperature sensor, etc.). Each device can be connected to only one gateway, and it can be of one preselected device type. Device type, on the other hand is assigned to either measurement or control type, depending on the set of supported functionalities (e.g. smart plug can measure power consumption but it can also perform a control action on the connected appliance in order to turn it on or off). Finally, for each device group, an information regarding its geographical location is stored, so that current and forecasted weather information can be

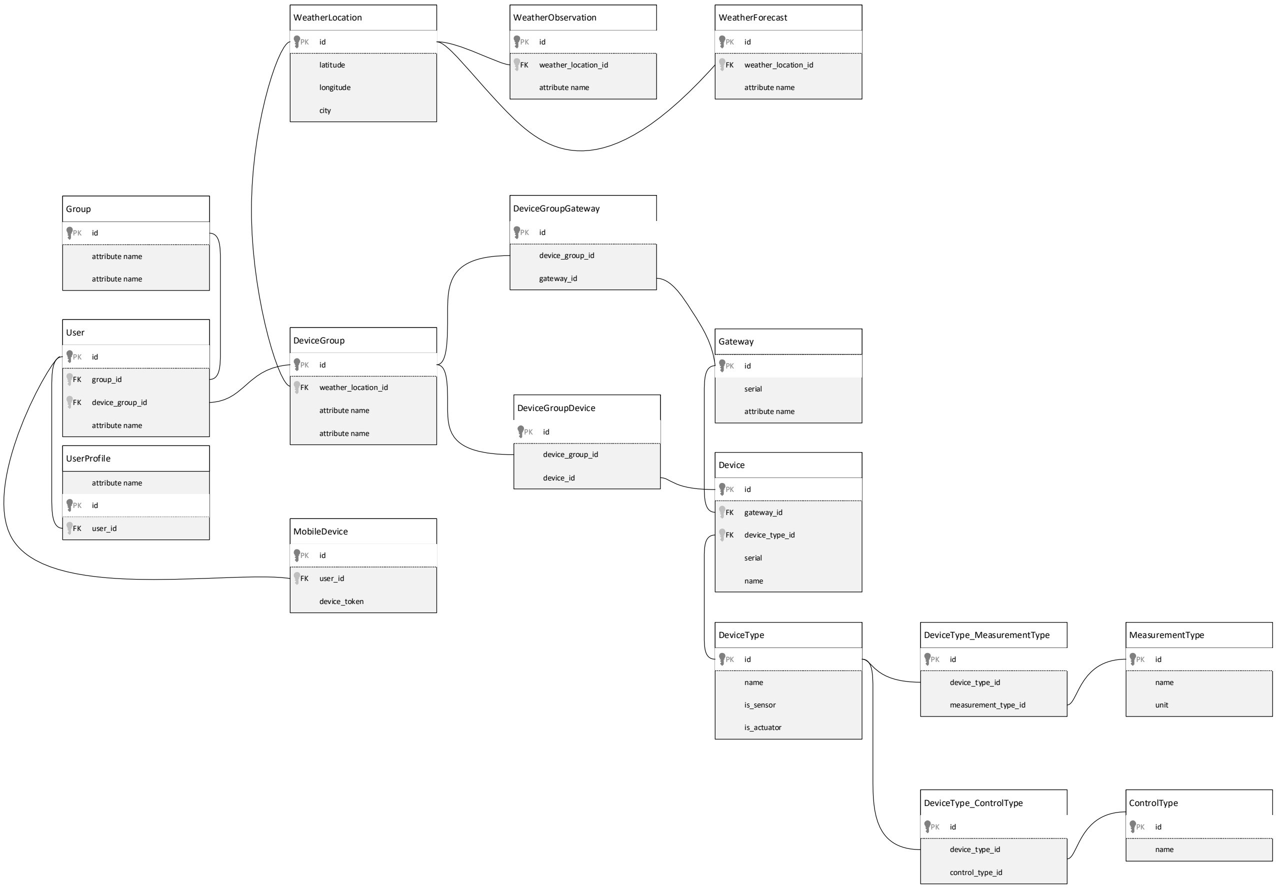

Figure 4 MySQL data model of SEM mobile application backend

fetched from the external weather service API, and subsequently shown to the users via SEM mobile application.

V. Conclusions

In this paper, we presented the Smart Energy Manager application developed as a visualization interface towards end users of InBetween platform. The cloud platform is based on open source software which is deployed with the goal to integrate the devices deployed in the field with the advanced energy services. SEM application, which is based on Android OS, allows end user to observe measured data, obtain suggestions and even perform some of the control actions on the deployed devices automatically.

Acknowledgment

This work was partly financed by the European Union (H2020 InBetween project, Pr. No: 768776), H2020 LAMBDA project, Pr. No: 809965) and by the Ministry of Science and Technological Development of Republic of Serbia (Pr. No: TR-32010).

References

[1] FP7 project EnergyWarden

[2] FP7 project EPIC-HUB

[3] Energic application www.energic.io

[4] Smappee application www.smappee.com

[5] eGreen www.public.egreen.fr

[6] Jarvinen H., and Vuorimaa P., "Semantic Interoperability for Web Services based Smart Home Systems," In Proceedings of the International Conference on Agents and Artificial Intelligence (ICAART-2015), pages141-148.

[7] Jiang, L., Liu, D.-Y., Yang, B., et al. (2004). Smart home research. In Proceedings of the Third Conference on Machine Learning and Cybernetics, pages 659–664.

[8] Capitaneli A., Papetti A., Peruzzini M., Germani M., "A smart home information management model for device interoperability simulation," at 24th CIRP Design Conference, Milan, Italy, 14–16 April, 2014, CIRP21 (2014) 64–69.

[9] T. Hargreaves, T. "Practicing behaviour change: Applying social practice theory to pro-environmental behaviour change." *Journal of Consumer Culture* vol 11 pp 79-99, March 2011.