Application of the open-source cloud platform FIWARE for future building energy management systems

T. Storek, J. Lohmöller, A. Kümpel, M. Baranski and D. Müller

RWTH Aachen University, E.ON Energy Research Center, Institute for Energy Efficient Buildings and Indoor Climate, Mathieustraße 10, 52074 Aachen, Germany

E-mail: `tstorek@eonerc.rwth-aachen.de`

Abstract. The integration of volatile, decentralized energy into building energy systems requires intelligent algorithms that are capable of controlling the augmented complexity and the interactions of subsystems efficiently. Not only do the algorithms require computing capacity, but also a feasible information and communication infrastructure. One possibility to provide computing power on demand is cloud computing that forms together with an extended connectivity of physical devices the Internet of Things (IoT). We present a customizable open-source IoT platform setup using FIWARE. For the integration of existing building automation networks, we extend an OpenMUC gateway with BACnet functionality and connect it to our platform via MQTT. We successfully demonstrate the overall setup with a web-based visualization of high resolution energy monitoring and a control loop as basic requirements for the integration of more advanced control algorithms in future building energy management systems. We find our setup generally feasible for both use cases and, hence, will aim for the integration of more advanced control algorithms in the future using the presented platform components.

1. Introduction

Nowadays, the fraction used for heating, ventilation, air conditioning and refrigeration (HVACR) accounts about two-thirds of the building energy consumption. Hence, there is an urgent need for energy saving strategies, which is also a focus of energy policies in many countries [1–3]. Optimizing the operation of building energy systems (BES) holds high potential for energy savings [4; 5]. However, the integration of volatile, decentralized energy sources and the coupling of the thermal and the electrical domain significantly increases the complexity of future BES. Hence, a reliable operation based on classical On/Off or proportional-integral-derivative (PID) control becomes more cumbersome and can lead to low energy efficiency. Therefore, there is a need for advanced control approaches like model predictive control (MPC) handling the increasing complexity. In the literature, energy savings of about 20 % to 35 % and reduced operating cost of up to 73 % are reported [1–3]. Nevertheless, the practical application of MPC is quite rare and still objective of ongoing research [1–3]. Usually, modelling and calibration are considered as most important and, simultaneously, most cost-intensive parts [1–3]. In practice, however, the communication between the control algorithms and the hardware level of building automation systems (BAS) and its specific protocols is still a challenging task that is often underestimated and represents a high hurdle. Consequently, only a few experts know how to set up and commission an MPC successfully [3]. Therefore, a simplified communication structure, as the idea of the internet of things (IoT) promises, is highly desirable. Additionally, many advanced control algorithms occasionally require significant computational power that is usually not available on site. Hence, it seems obvious to run them on an external scalable cloud system in order to save local resources. Today, many commercial IoT platforms are available. Among the best-known products are Amazon's AWS IoT that offer IoT-Services for industries, home automation and the commercial sector or Microsofts's Azure IoT that extends the portfolio of services to the energy and security sector. Non-residential buildings and their BES are usually planned highly individually, with many distributed sensors and actuators. Therefore, they seem to be an ideal application for IoT. However, to the best of our knowledge, there are only a few commercial solutions for newly constructed buildings as well as for the integration of the building stock and its BAS. One possible reason is that the procurement system for construction sites is leading to a fragmented marked and high pressure on pricing [5]. Due to higher price levels other automation sectors, e.g. plant automation, seem to be more attractive for developing IoT business cases at the moment . Furthermore, the individuality of buildings requires highly adaptable solutions with a high degree of interoperability as, otherwise, the customer will be forced into an unacceptable vendor-lock.

Hence, in this work we present a customizable open-source cloud-platform for exploiting the advantages of the latest developments in cloud technologies for smart building energy management systems (BEMS) and a gateway to realize the communication between existing BEMS and the platform. Furthermore, we investigate two domain-specific use cases for proving the setup's feasibility for future building automation applications.

2. Platform development

As central data storage and management system, we make use of FIWARE, which is a framework of open-source platform components created to facilitate the development of smart solutions within various application domains [6]. To integrate our platform with existing field layers of BEMS, we employ the open-source framework OpenMUC [7] as BAS network interface and modify it for our needs.

2.1. FIWARE IoT platform

Beside the fact that FIWARE is freely distributed, it comes along with the benefits of a large community and offers a set of advanced components including a high performance database engine and a sophisticated set of Representational State Transfer (REST) application programming interfaces (API) using the standardized Next Generation Service Interface (NGSI) format, which is also the formal standard for context information management systems in smart cities [6]. Thus, it is promising for future application in BEMS. At the moment, the FIWARE catalogue contains about 30 interoperable software modules, so-called Generic Enablers (GE) for developing and providing customized IoT platform solutions [6]. This work only employs some core modules of the overall framework in order to proof FIWARE's general feasibility for BAS applications, leaving aside additional GEs, such as the components for user identity management.

Figure 1 depicts the architecture of the deployed IoT platform prototype. The functionality of the deployed GEs is outlined in the following. The Orion context broker is the central component of the FIWARE stack that provides update, query, registration and subscription functionality via its API for managing the context information in the platform. The data itself is stored in an underlying MongoDB database [9]. Orion only provides information of the current data content. Hence, for storing time series data persistently, we deploy QuantumLeap, which subscribes a specified content and automatically stores updated data persistently in the connected high-

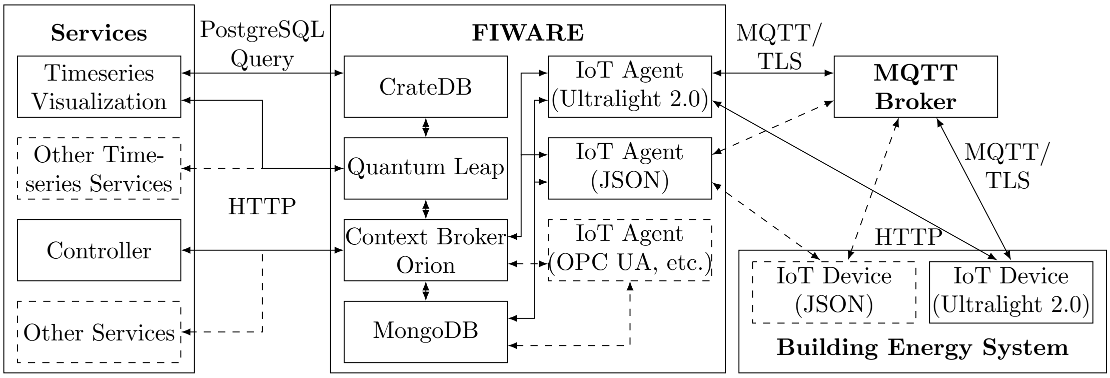

Figure 1. IoT platform architecture using the FIWARE framework as context management system receiving data from IoT devices via a MQTT Broker and providing it to BEMS cloud services for analysis and operation.

performance CrateDB database [6; 10]. Similar to Orion, QuantumLeap provides an API for managing the historic data stored in the database and, thus, adds time series functionality to the platform. Via the two APIs, data can be provided to any external service such as visualization, analysis or control algorithms. For seamlessly connecting, managing and gathering data of IoT devices, FIWARE offers a set of IoT Agents that translate IoT specific protocols and message formats, such as Ultralight 2.0, JSON, etc. into the platform-internal NGSI format. Devices located in the building energy system send and receive the data either directly via HTTP or via an additional Message Queueing and Telemetry Transport (MQTT) Broker. Particularly, in this paper, we use the open source broker implementation of Eclipse Mosquitto [11]. The latter also provides authentication and Transport Layer Security (TLS) encryption mechanisms. Whenever a device is registered at an IoT Agent via its API, the agent automatically connects the device and the corresponding data with a specified content in Orion and stores the configuration in the MongoDB. Hence, each time the device measurement is updated, the data of the corresponding content in Orion is instantaneously updated; conversely, if a command in a content is updated, it is directly sent to the device.

To exploit the hardware potential of cloud technology, we embed our platform prototype into a container virtualization using images provided by FIWARE. Not only does this enable an easy setup and configuration procedure, but also the distribution on multiple hardware nodes via Docker-Swarm [13]. Furthermore, in order to facilitate the handling of the APIs and accelerate the development of services, we developed a FIWARE library for Python (FiLiP). The library as well as the configuration files of presented platform setup is available at `https://github.com/RWTH-EBC/FIWARE`.

2.2. Building Automation Network Interface

In BAS, the Building Automation and Control Networks (BACnet) is a standard protocol on the field and automation layer [5]. To enable communication between the FIWARE services and BACnet devices, we deploy the open-source framework OpenMUC [7] as central gateway software. For this purpose we update an existing OpenMUC-driver[1] for BACnet. In particular, we use a recent release of BACnet4J [8] and apply minor modifications. We increase compatibility with different device vendors, improve the performance on larger BACnet installations and query additional object attributes useful to distinguish similar objects. With

[1] available at `https://github.com/openmucextensions/bacnet`

our modifications, we are able to automatically scan and register hundreds of BACnet devices providing thousands of data objects in non-residential buildings within only few minutes. Depending on a device's capabilities and response time, OpenMUC either samples data using a fixed rate or subscribes to value changes.

For secure and bidirectional communication with the FIWARE platform, we employ the MQTT protocol. More precisely, we add a connector service to OpenMUC that publishes value changes to the FIWARE platform immediately. All data is sent over a single, persistent MQTT connection established between the gateway and a MQTT broker. This architecture allows gateways located at remote buildings to be hidden behind firewalls, i.e., it only requires the MQTT broker to be accessible from a remote building network. Furthermore, no FIWARE services need to be exposed to remote networks.

3. Building Monitoring

The platform setup provides the basic functionality for IoT applications, yet, it does not provide any specific modules for the building energy systems. Hence, to proof the automatic integration and logging of multiple devices from BAS, we demonstrate a web-based visualization of high-resolution energy monitoring of a large office building.

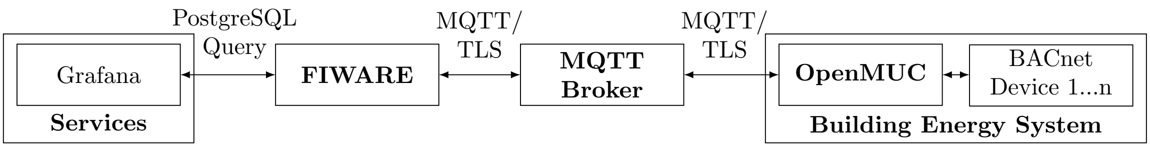

Figure 2. Platform setup for realizing a high-resolution building monitoring of an office-building with BACnet based field layer.

The particular BES is equipped with a combined heat power (CHP) engine, two condensing boilers and a heat pump (HP). The automation and field layer uses BACnet as primary communication protocol. The BEMS delivers more than 7300 data objects provided by hundreds of sensors and actuators. For automatic BACnet integration and logging of all data points, we deploy the presented OpenMUC gateway and realize communication to the cloud over MQTT with TLS encryption. The setup is illustrated in Figure 2. By an automated network scan, OpenMUC creates an internal list of devices, data points and available meta data. Parsing this list using the developed Python Library FiLiP, we automatically register all data points at the IoT-Agent for Ultralight and initiate the logging of the corresponding content in Orion via the QuantumLeap API. For visualization of time series data, we deploy the open source web visualization framework Grafana [14] retrieving the data directly over the PostgreSQL interface of the CrateDB. Nevertheless, for more sophisticated setups, the functionality for retrieving time series over the QuantumLeap API should be used.

Due to many different parties within BES, naming schemes and and data structure tend to be inconsistent and vendor-depending. Although the integration process of devices works fine parsing the network data point list from OpenMUC, we find the data structure of existing BAS represents a mayor hurdle. Hence, the information can only be accessed if the exact name of a data point is known. Additionally, common naming schemes do not provide any semantic information, which makes it cumbersome to interpret data sets and also to process them automatically and run analysis algorithms. Therefore, a generalized naming scheme such as the BUDO schema as suggested by Stinner et al. [15] is highly desirable in future BAS. Furthermore, in future work we will extend the implemented parser software by including additional BAS-specific context information sources, such as engineering data exchange tables in order to obtain a semantically structured building monitoring.

4. Building Control

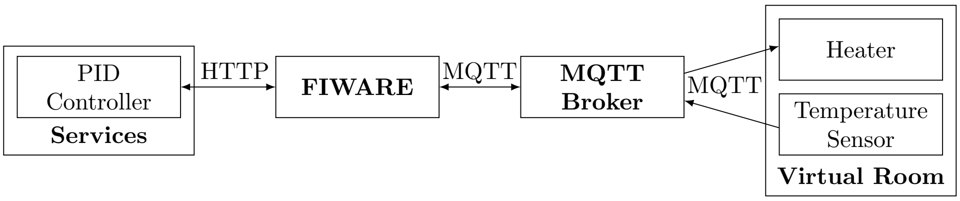

Figure 3. Software in the loop setup for demonstrating a closed control loop using FIWARE.

To investigate sending and receiving control commands through the platform, we implement a closed-loop control in the software-in-the-loop setup illustrated in figure 3. The setup consists of the FIWARE platform, handling all data content and flows, an MQTT broker for communication, a PID controller implemented in Python and virtual room. For the latter, we use a functional mock-up (FMU) a thermal zone model from the open source Modelica library AixLib[2] [16]. Using FiLiP, we integrate the zone temperature sensor and an ideal zone heater as devices to our platform. Whenever the FMU receives a command, it conducts a fixed simulation time step. The sample based controller is connected via the Orion API, where it retrieves virtual sensor measurements generated by the FMU and returns its calculated control signal to the simulated heater. Running the simulation for a winter day using the test reference year 2012 (TRY2012) of San Francisco, we obtain the results depicted in figure 4. By decreasing the sample rate of controller actions, we are able to send several commands per seconds before we experience latency effects, which mainly occur due to the computational time required for simulation. Nevertheless, due to the physical inertia of real BES, we are confident that the platform is capable of to reliably transmitting control signals in a BAS.

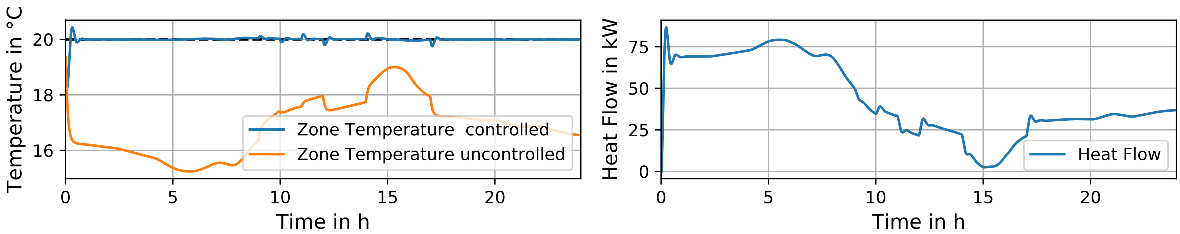

Figure 4. Trajectory of a thermal zone temperature for a winter day with and without control (left). Controlled heat flow of an ideal heater (right).

5. Conclusion and future work

This paper presented a prototype of an IoT cloud platform for future building energy management systems (BEMS) based on the open source FIWARE framework. Embedding the required FIWARE modules into a container virtualization leads to an easy and scalable platform setup. For facilitating the handling of the platform APIs, we presented a FIWARE library for Python (FiLiP). Extending the open source gateway framework OpenMUC with MQTT and BACnet functionality assures the integration of existing building automation systems into the platform. Deploying the platform setup together with the gateway for a building monitoring, we found that the automated registration and logging of more than 7300 data sources worked

[2] Available at: `https://github.com/RWTH-EBC/AixLib`

seamlessly. Grafana represents an easy-to-use web visualization of the logged data. Additionally, we demonstrateed a fully functional closed PID control loop in a software-in-the -oop setup with FIWARE as central communication platform, where the platform proves its reading and writing functionality. From the successful demonstration, we draw the conclusion that FIWARE, combined with a gateway, is a suitable and customizable, low-cost but yet powerful solution for future BEMS. Future work will focus on the integration of IoT devices and automatically adding semantic information as well as the implementation of advanced control algorithm and their demonstration applying the presented platform. All source code and examples as well as the required setup files are publicly available at `https://github.com/RWTH-EBC/FIWARE`

Acknowledgments
We gratefully acknowledge the financial support provided by the Federal Ministry for Economic Affairs and Energy (BMWi), promotional reference 03ET1561B.

References
[1] Afram A and Janabi-Sharifi F 2014 *Building and Environment* **72** 343–355 ISSN 0360-1323
[2] Baldi S, Michailidis I, Ravanis C and Kosmatopoulos E B 2015 *Applied Energy* **154** 829–841 ISSN 0306-2619
[3] Killian M and Kozek M 2018 *Energy and Buildings* **158** 1404–1416 ISSN 0378-7788
[4] Fütterer J P, Schild T P and Müller D 2017 Gebäudeautomationssysteme in der praxis URL `http://publications.rwth-aachen.de/record/691150`
[5] MeGA 2016 Marktstudie 2016 gebääudeautomation schweiz www.mega-planer.ch URL `https://www.mega-planer.ch/fadaladdondlz/files/.addonpublikationeintragfile/publikationen/90.pdf/MeGA_Marktstudie_GA_2016.pdf`
[6] FIWARE Foundation, eV 2019 FIWARE: The open source platform for our smart digital future accessed: 2019-05-25 URL `https://www.fiware.org/about-us/`
[7] Feuerhahn S, Zillgith M, Becker R and Wittwer C 2009 *ETG-Kongress* (Berlin: VDE-Verlag) pp 177–180
[8] Infinite Automation Systems 2017 BACnet4J - Open Source BACnet Library accessed: 2019-05-25 URL `https://infiniteautomation.com/bacnet4j-open-source-bacnet-library`
[9] MongoDB, Inc 2019 mongodb accessed: 2019-05-25 URL `https://www.mongodb.com`
[10] Crateio, Inc 2019 cratedb accessed: 2019-05-25 URL `https://crate.io`
[11] Foundation E 2019 Eclipse mosquitto™ - an open source mqtt broker accessed: 2019-05-25 URL `https://mosquitto.org/`
[12] Merkel D 2014 *Linux J.* **2014** ISSN 1075-3583 URL `http://dl.acm.org/citation.cfm?id=2600239.2600241`
[13] Soppelsa F and Kaewkasi C 2017 *Native Docker Clustering with Swarm* (Packt Publishing) ISBN 1786469758, 9781786469755
[14] Labs G 2019 The open platform for beautiful analytics and monitoring accessed: 2019-05-25 URL `https://grafana.com/`
[15] Stinner F, Kornas A, Baranski M and Müller D 2018 *The REHVA European HVAC Journal - August 2018* The REHVA European HVAC Journal ed REHVA pp 10–15 URL `https://www.rehva.eu/fileadmin/user_upload/10-15_RJ1804_WEB.pdf`
[16] Müller D, Lauster M, Constantin A, Fuchs M and Remmen P 2016 *BauSIM 2016: Sixth German-Austrian IBPSA Conference* pp 3–9