Centralised monitoring and control of buildings using open standards

Pavel Paulau[1], Johannes Hurka[2], Jan Middelberg[2], Sascha Koch[1]

[1] Institute for Applied Photogrammetry and Geoinformatics (IAPG),
Jade University of Applied Sciences Wilhelmshaven/Oldenburg/Elsfleth, Ofener Str. 16/19, 26121 Oldenburg
pavel.paulau@jade-hs.de
[2] Jade University of Applied Sciences Wilhelmshaven/Oldenburg/Elsfleth, Ofener Str. 16/19, 26121 Oldenburg

Keywords: building control, district level, sensors, actuators, open standards, OGC Sensorthings API

Abstract

Due to the necessary heat turnaround, buildings are increasingly seen as part of districts, e. g. Positive Energy Districts. Optimizing the energy efficiency of buildings is an essential part of implementing the heat turnaround. As a result, sensor/actuator systems are becoming increasingly important, both at building level and district level. Various proprietary and open standards for monitoring and control are used at building level. This heterogeneity makes it difficult to jointly optimize energy efficiency in buildings. OGC Sensorthings API (STA) is an open standard that defines a common language for sensor data and even tasking processes. In addition, KNX and LoRaWAN are available as standards for building monitoring and control. This article therefore proposes a solution that combines these standards. As proof of concept, two different buildings are equipped with KNX and LoRaWAN. Sensor data is collected in both buildings and merged centrally in a so-called building data lake based on OGC STA Sensing Part, e.g. temperature or indoor air quality. In addition, OGC STA Tasking Part is used to trigger actuator commands centrally and independently of the communication standard used in the controlled building, e. g. temperature setting of thermostats. As part of the proof of concept, it is shown how standardized sensor data can be visualized based on available visualization software, e. g. Grafana, and how the standardized actuator control level can be used for optimizations such as the night setback of heating systems.

1. Introduction

Our overall goal is to use digitalization to minimize the energy consumption of buildings and related CO_2 emissions to reach the exit from nuclear and fossil-fuel energy consumption. This requires the use of complex systems based on photo voltaic power stations and heat pumps, which must work synchronously. The "Energy Efficiency Principle" of EU (Team of EU parlament and council, 2023) imposes new requirements on the governments of countries. It results in more complex installations and control systems in buildings and requires reliable and transparent management of heterogeneous (sensor) data. An even bigger potential for efficiency gains lies in the common management of many buildings, eg. Positive Energy Districts (Zhang et al., 2021, Lindholm et al., 2021) and Smart Cities.

The most important use cases in our work are the collection and storage of measurements for monitoring and analysis of building physics and the implementation of an interface for controlling the technical systems of the buildings. Our more ambitious aim however is to control the actuators using self-learning algorithms (Wang and Hong, 2020). In order to develop universally applicable solutions, it should be avoided to depend on the interface technology and nomenclature of single manufacturers or products.

Therefore, we are working on a software design towards a scalable and interoperable data infrastructure for sensor data, actuator commands, and metadata concerning the buildings and their technical systems. Different aspects of interoperability (for building a system of systems) are discussed in (Atkinson et al., 2022, Chaturvedi and Kolbe, 2019). The aim of our article is a concept to organise the integrative data acquisition and control in buildings according to the standard SensorThings API (STA) of the Open Geospatial Consortium (OGC): (OGC Team, 2021), (OGC Team, 2019), (OGC Team, 2022). According to the standard, the OGC SensorThings API provides an open, geospatial-enabled and unified way to interconnect the Internet of Things (IoT) devices, data, and applications. The Sensing part provides a standard way to manage and retrieve observations and metadata from heterogeneous IoT sensor systems. The Tasking part provides a standard way for the parameterizing of taskable IoT devices, such as individual sensors and actuators.

One of the subjects of the conference "Smart Data & Smart Cities 2024" (UDMS Team, 2024) is "open urban platforms". The word open could mean one of following:

1) The data of platforms are open for everyone.
2) The software behind the platforms is open source.
3) The platforms are implemented with open standards.

The cases 1) and 2) are very nicely elucidated in (Happ and Wielgosch, 2023) including the discussion, which data are reasonable to be open for all, which data even "must be open" via public sector organisations according to EU guidelines. The paper contains also the examples of implemented open urban platforms (mainly in Germany).

The use of the STA should satisfy all three points. Some examples of usage can be found in (Emde et al., 2022, van der Schaaf et al., 2020) for environmental informational systems, in (OGC Team, 2022) for Citizen Science, in (Fischer et al., 2021) for the networking of digital spaces.

We deal however not only with sensors but also with actuators. In this respect it is questionable whether there are cases where public control of devices is acceptable. Furthermore, data of

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-4-W4-2024-169-2024

some sensors (eg CO_2-concentration in room, hence information about the time resolved presence of a persons in the room) are related to individuals if combined with data sources such as official web pages where the names of staff members and their office numbers can be found. The individuals do not necessarily wish to make it open. Also, we must work using compliance from people, according to General Data Protection Regulation GDPR (European Union, 2024). It follows that we can have an "open platform" only in sense 2) and 3) but not 1).

An example of a hardware/software system is the Smart Meter Gateways (SMGW) and the corresponding software, which is becoming the central system to monitor the renewable power plants and to control heat pumps in buildings. Finely tuned access restrictions must be implemented there to handle data about the energy consumption of customers. The SMGW are implemented according to the guiding principles (BSI Team, 2024). These guiding principles however contain significant complexity and much effort is required to enter the field from the scratch. Although the guiding principles of SMGW are open, there are not yet open source tools, which would implement the corresponding guiding principles.

A number of technologies are currently used for data acquisition in buildings, for example SigFox, LoRaWAN, NarrowBand IoT, MIOTY, LTE-M, BTE, Wireless-MBus, M-Bus, KNX (Orfanos et al., 2023), when only small amounts of data must be sent and received by devices, which has the big advantage of low power consumption. Additionally, the usual technologies as WLAN, LAN can be used. STA allows to integrate all these technologies (both for sensors and for actuators), thus providing a common interface.

The research results presented are based on design science research (Hevner et al., 2004). The relevant steps are therefore the definition of objectives for a problem solution, design and development as well as demonstration and evaluation. Prototyping is used as an evaluation pattern for design science research (Sonnenberg and vom Brocke, 2012).

In the next section, we will describe two buildings with different energetic properties that are equipped with different technologies but are connected to the same data infrastructure. The buildings are needed for demonstration and evaluation.

The centralised approach to monitoring and controlling these two buildings (and in principle many more) with OGC STA and the resulting data architecture are designed in section 3. The two main applications that depend on the data infrastructure are shortly described in the section 4. Building control and visualization of building data using OGC STA Sensing and Tasking is demonstrated and evaluated. Finally, the section 5 with the conclusions follows.

2. Building digitalisation and automation

For our research, building 1 and building 2 are equipped with different digital devices and are used to perform different experiments.

2.1 Building 1. Automation with KNX.

Building 1 is new and was built according to very high insulation standards as a test bench building for fossil-free heating and for study courses and seminars. It covers about 50 m^2 ground floor area and an additional 21 m^2 on the first floor as an open gallery. The specific transmission heat loss is $H_T = 72\frac{W}{K}$. The building has a 10 kW Peak photovoltaic installation on the roof and a heat pump as well as electrical (battery) and heated water (tank) energy storage (see Figure 1). An air convector with ventilator can extract the energy from the circulating water coming from the heat pump or the energy storage tank with very high power. Additionally, solar heating through the large south oriented window is part of the heating concept.

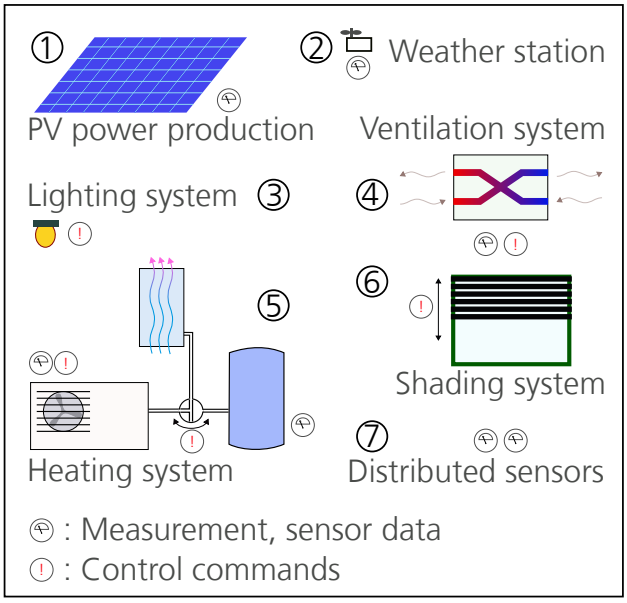

Figure 1. Overview of the technical systems in building 1 which are connected via KNX. 1: 10 kW Photovoltaics system with energy meters. 2: Weather station with sensors for irradiation, wind speed, temperature, rain. 3: Lights with dimming and switching actuators. 4: Ventilation with heat recovery, sensor and status data, different controls. 5: Heat pump, convector, heat storage with controllable 3-way valve. 6: Controllable shading for windows. 7: Sensors for temperature, relative humidity, presence, light, CO_2, and window status distributed throughout the building.

In this building, the open standard KNX (ISO/IEC Team, 2006) is extensively used for communication between the technical components. KNX is widely used for energy management solutions (Tee et al., 2023), (Muñiz et al., 2024) and for connected districts (Sita and Dobra, 2014). It uses a dedicated twisted pair bus for communication of the devices, which come from many manufacturers. In a typical KNX installation, there is no central node. Instead, each device is configured to send to and/or listen to specific group addresses. Thereby, a sensor event in one device can lead to an action of another device. However, we also use a small central server which allows for visualisation and app control, as a gateway to integrate the KNX devices with our central data infrastructure.

One of the objectives of our research is to find strategies that make maximal use of renewable energy via operation of the heat pump to fill the water energy storage or to heat the building directly.

2.2 Building 2. Automation using LoRaWAN

In building 2 we basically implement the same concept as in building 1. It is an older building and has not very high insula-

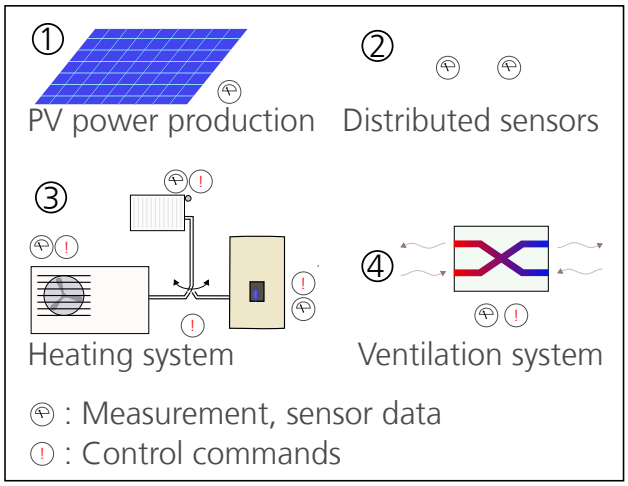

Figure 2. Overview of the technical systems in building 2 which are connected mostly via LoRaWAN. 1: Photovoltaics system with energy meters. 2: Sensors for temperature, relative humidity, presence, light, CO_2, and window status distributed throughout the building. 3: Bivalent heating system with heat pump and natural gas boiler, energy meters, radiators with remotely controllable thermostat valves. 4: Ventilation with heat recovery, sensor and status data, different controls.

tion standard. It has 3 levels, around 21 workplaces in around 15 offices. Therefore, the operation only through heat pump would require very high power of the heat pump in the coldest winter months. In this case, it is more reasonable to combine the heat pump with the already existing conventional gas heating (see Figure 2).

Accordingly, the control system has to switch between these two heat sources in order to achieve the minimal use of fossil energy. Also, the room temperature in unoccupied rooms should be reduced automatically. Accordingly, a network of radio controlled LoRaWAN (Krishnan, 2020, LoRa Alliance, 2016, Happ and Wielgosch, 2023) thermostat valves and motion detectors were installed.

3. From smart home to smart city

The two buildings do not in principle have to be controllable via a common server, because they could be controlled via their own local systems or servers. We decided however to control the buildings with a central server, which opens the ways to work also on the district or city level. For example, the current production of renewable energy from PV installations on the roofs of different buildings can be used as basis for decisions to reduce or to increase the power of heat pumps in separate buildings, or to optimise the usage of stored heat based on global city information and not only on local building information.

3.1 Software architecture

Our software architecture is shown in Figure 3, where the two buildings are represented with text labels in the bottom of the figure. The devices in these buildings are connected to the central FROST server via different media. Some of them also need special gateways. With FROST (Fraunhofer IOSB Team, n.d.), an open source implementation of the SensorThings API was chosen which will be described in section 3.3. The boxes ”Cable based/KNX” and ”Radio based/LoRaWAN” denote the networks of sensors and actuators, which are installed in corresponding buildings. The box ”C-Script/Logger (i2c-bus, serial) WLAN” represents an additional network of sensors in building 1. These sensors communicate directly with the central FROST server over WiFi. Another network of sensors ”Python Script/Logger (i2C bus, serial) LAN” is installed in building 2 and directly connected via LAN.

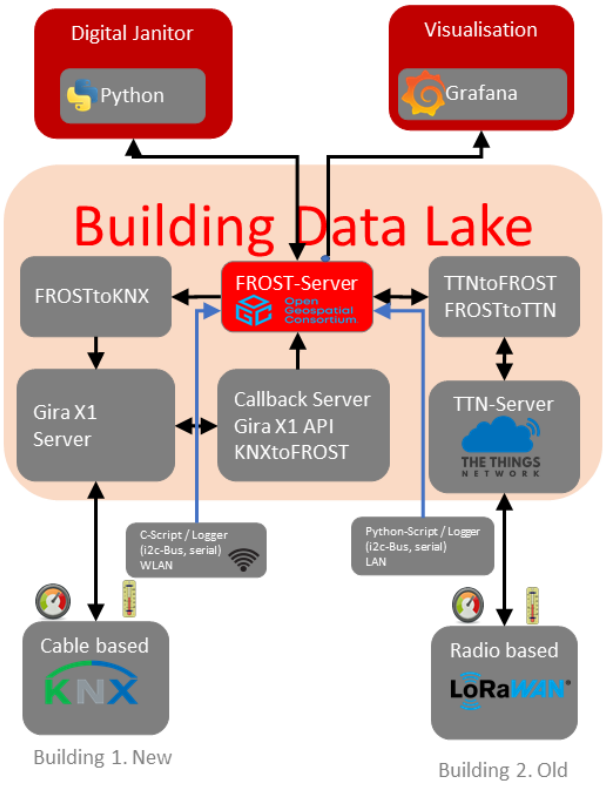

Figure 3. Software Architecture. Arrows show the direction of data flow. The blue arrows denote direct communication between sensors and central server.

The LoRaWAN devices have manufacturer provided integration with ”The Things Network” (TTN) server over the network of the LoRaWAN gateways. The sensors/actuators communicate with the TTN server as shown with corresponding arrow. The TTN Server has no direct integration with FROST, therefore we have built such integration, shown as the gray box ”TTNtoFROST/FROSTtoTTN” (for more details see section 3.4.2).

The KNX devices in the other building also need a gateway and special software for the integration with FROST, described in more detail in section 3.4.1.

The box ”Digital Janitor/Python” in Figure 3 refers to a computer program which takes the sensor data as input and returns the new states of all actuators as output. It is intended to use machine learning to minimise energy consumption and to maximise the comfort of people. Currently, and for the aims of this paper, the Digital Janitor is implemented using hard coded rules (see Section. 4.1). The Box ”Visualisation/Grafana” in the software architecture denotes the software used to visualise the data for monitoring (see section 4.2).

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-4-W4-2024-169-2024 | © Author(s) 2024. CC BY 4.0 License.

3.2 Building Data Lake

According to the definition (Fang, 2015, Mathis, 2017): *"An (enterprise) Data Lake is a Big-Data-oriented keeping of data, where the big amount of a poly-structured (structured, semi-structured and unstructured) raw data are collected during the long time from different sources in corresponding source formats and application neutral.* To implement our tasks, we perform data acquisition using different tools: AutoCAD documentation, PDF datasheets and descriptions of measurement concepts, Smartphone Photos. Some tools export the information in proprietary file formats. This additional information is not trivially translatable into the simple texts and numbers supported by the FROST database. Therefore, such a documentation must be referenced from the database. All these files are part of our data lake and belong to the conception of the measurement and control experiments, implemented using the FROST server. Nevertheless, we will mainly discuss here the sensor/actuator data acquisition part of the building data lake, which is the most important for the aims of this paper.

3.3 FROST system

To store the data from sensors and actuators a DBMS is needed. For create, read, update and delete (CRUD) operations over the internet, an interface between the DBMS and the client is necessary. The FROST server (Fraunhofer IOSB Team, n.d.) combines postgreSQL as DBMS with support of Structured Query Language (SQL), mosquitto as a broker for the Message Queuing Telemetry Transport (MQTT) and a web server as hypertext transfer protocol (http) interface between the client and DBMS to implement the necessary functionality.

The CRUD operations are implemented via representative state application programming interface (REST API) as http (Get, Post, Patch and Delete) requests to the web server. One needs to provide the arguments to the post and patch methods in JSON format according to the standard (OGC Team, 2021), (OGC Team, 2019). Mosquitto MQTT publishes messages to implement the live transfer of data and commands. MQTT publishes the changes in all tables in the database, so that the clients immediately receive this information. The changes in the observation table are relevant for monitoring the state of the building. The publication of changes in the tasking tables of the database allows the transmission of control commands to actuators. The MQTT server also provides the function to write the measurements of the sensors into the data base.

3.3.1 Data base management system. Data schema The basics of database management systems and relational data bases are presented in (Jarosch, 2016) including structured query language. SQL has passed the long evolution, starting from the first standard via ISO (International Standardization Organization) in 1986 and with actual standard from 2023 (ISO Team, 2023). According to the concept of a DBMS, the requests to perform operations with data should be unified, the software behind the DBMS should be exchangeable. The same idea to use unified language has been further developed in OGC STA standard (OGC Team, 2021). While the SQL standard does not define content restrictions on data, the OGC STA however imposes a fixed number of tables in the database schema, as well as names of the tables, their internal structure (types of columns) and fixed connections between the tables.

The data schema of STA (including the Tasking Core) with the explanation of its usage for our use case is shown in Figure 4. Many modern sensoring devices combine multiple sensors for different measurements, for example (Elsys Team, 2020). Sometimes, there is no detailed information about the sensors of a multi sensor in the data sheets. Some sensors are also combined with actuators (MClimate Team, 2021). For these reasons, the table "Sensor" is not exactly suitable to describe such a device. We decided therefore to use the table "Things" to describe the devices with as much entities of "Datastream" as there are sensors within the device. Each datastream is connected to one of the entities of "ObservedProperty", which describes the type of measurement, eg. temperature or CO_2 concentration. Thereby, ambiguities related to differently named sensor channels in different devices can be avoided. Each measurement creates an entity of type "Observation", connected to the appropriate "Datastream". For tasking, we use the table "Actuator" to store the type of the device, eg thermostat valve "Vicki". In the table "TaskingCapability" each entity represents an instance of such an actuator which is connected to the corresponding "Thing" and to the table "Task", where the multiparametric commands are stored, which where sent to the actuator. The STA does not provide tables to specify "building" and "room", therefore we use the table "Location" to describe rooms, with information about the building in the description and properties. All "Things" belonging to a room are connected to the "Location". If commercially available devices

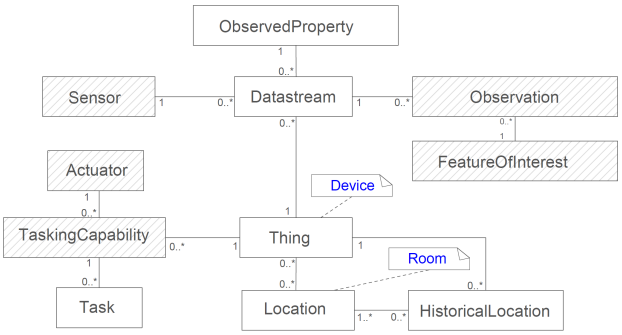

Figure 4. STA data schema with tasking core and explanations of its application for the use case described in this paper. The shaded tables are currently not used but initialised with dummy entities where necessary.

are used, there will often be several instances of the same type of device. To avoid redundancy and strengthen reusability of data, it would be practical to have a table for the type of device and one for the instances. But there is only one table "Thing". There, we store the data about instances as entities, and define the type of thing as property in the record. Only the short name without any additional information (like link to documentation, link to payload decoder) related to the device is stored.

The structure of the STA data base assumes the existence of unique IDs, which are generated by the system and are returned to the clients at create operations. In the course of manually creating entities, it is however not always practical to keep track of these IDs, to be able to later find the just created entity. For our use case we added a naming convention, which is not part of STA standard: the names of entities must be user defined and unique.

- Location name: A.B.C.
- Thing name: A.B.C.D.E.
- Datastream name: A.B.C.D.E.F.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-4-W4-2024-169-2024 | © Author(s) 2024. CC BY 4.0 License.

Each capital symbol can be a word (without use of points). A - represents the city, B - the building, C - the room, D - is optional and denotes the number of the device within the same room, E - represents the type of the device, F - represents the name of the measured property.

3.3.2 FROST web and MQTT servers For security reasons, it is strongly recommended not to open the data base servers for the use from the internet. The FROST server provides a web server which supports HTTP requests written according to STA, translates them into SQL requests and redirects the requests to the database server, thus providing CRUD operations for clients outside the local network. The tasking requires not only to save the data to the database, but also to inform the clients waiting for commands. The commands are transmitted over MQTT to all subscribers. Additionally, some applications may need the information about the change of some sensor data, which is also supported by publication of the according messages over MQTT.

3.4 Gateway services

To operate the gateway redirecting the data from some system to FROST and back, one needs to filter all the Things (devices) according to the following points:

1) system the gateway service is designed for.
2) data format of device the gateway service has to process.
3) direction of communication.

To solve the first point, we have introduced the property ”COMMUNICATION_TYPE”. It can have the values ”KNX”, ”LoRaWAN” or ”W(LAN)”.

For the second point, a property ”DEVICE_TYPE” is used, which is initialised with the model name of the device. The gateway service must accordingly provide functions to translate the data from the data format provided by the manufacturer to the data format agreed to be used in FROST.

To solve the third point, we filter all device types which are implemented in the corresponding gateway service.

In principle, the operation of the gateway services is not independent of manufacturer. The service must know how to communicate with each type of device. The control (monitoring) over STA is currently also not independent of manufacturer. An extension towards a manufacturer independent standard would be a definition of very detailed interfaces for different types of devices, like ”thermostat valve” with appropriate functionality. Then, the different models of devices could be controlled (monitored) really independent on manufacturer.

3.4.1 KNX - FROST KNX to FROST. In order to transfer the desired information from KNX devices to the FROST server, we use the Gira X1 server and its ”Gira IoT REST API” (Gira Giersiepen GmbH &Co. KG, 2020). The server provides a large set of predefined functions that can be connected to KNX group addresses and further parameterised with a graphical configuration tool. Each configured function gets an internally unique ID. The API provides the possibility to register value callback servers which will in turn receive any event in JSON format with the ID of the function and the value of the event. A small server written in Python and running on a single board computer registers itself upon startup with the X1 and receives the JSON data, matches the function ID with the corresponding data stream and sends the data as observation to the FROST server.

FROST to KNX. Another Python service implements an MQTT client listening to the FROST server. When a Tasking telegram intended for a KNX actuator is being received, the program looks up the matching ID of the corresponding X1 function and sends a HTTPS PUT request to the Gira IoT REST API.

3.4.2 TTN - FROST The TTN Server must be properly configured. The application is created, all the LoRaWAN devices entered into application and configured. The access key for application must be created and provided to the services, described below. The MQTT integration of the TTN server must be activated.

TTN to FROST. Sensing. We have written a python program using paho.mqtt library. The program initialises a MQTT client, which listens to the MQTT server of TTN. The list of topics, which are added to the submissions of the MQTT client is taken from the FROST server by filtering out all Things, which have property ”COMMUNICATION_TYPE” equal ”LoRaWAN”. As soon as the message is published in the MQTT Server of TTN, the content is transformed into STA and submitted to the web server of FROST.

FROST to TTN. Tasking. A python program based on the paho.mqtt library connects to the MQTT server of FROST and subscribes to a list of topics. The list of topics is based on the list of Things of the FROST server, which have property ”COMMUNICATION_TYPE” equal ”LoRaWAN” and which device types (”DEVICE_TYPE”) are supported by the service. As soon as a message is published by the MQTT Server of FROST, the content is transformed into the format understandable for recipient and submitted to the MQTT of TTN.

3.5 Privacy and data security

Although the STA standard (OGC Team, 2021) does not specify any mechanisms for authentication, the FROST implementation of STA has at least the mechanism of basic authentication also with the possibility to have read, write and admin roles.

Whereas the dominant use of the OGC SensorThings data model (and API) can be coined with the use case “single authority provides sensor readings to consumers”, in Citizen Science there are many contributors (citizens) that – together – create the big “picture” with their observations (OGC Team, 2022). That implies, that all contributors need write and read rights, at least to be able to find the Datastreams, where they want to contribute to. The absence of a granular rights management means that all contributors can read and write everything in the database. It means practically, that Single Authority consists of many contributors, which makes it already difficult to manage. Additionally, all consumers with reading rights are automatically allowed to read everything. These problems lead to the idea of a new extension to the STA standard (OGC Team, 2022), (OGC Team, 2023) based on requirements from Citizen Science.

One of the STA aims is reusability of data. It means, that the potential of relational databases must be used as much as possible. If all data belong to one relational database, then one can reuse the data very efficiently. If the complete set of data is splitted between owners, then some previously shared data

cannot not be shared anymore and appear as copies in the sets of the different owners. Such a copy will not be reusable anymore, but there will be privacy in the data. In the STAplus draft, the authors try to combine privacy and reusability as much as possible. The content of many tables is splitted between the users, some tables however become shared between all users. For example, the tables Location, Sensor, ObservedProperty, FeatureOfInterest in the STAplus draft are shared between all users. In the Location table, for example, the information regarding single buildings could be stored but it must not be publicly available.

Regarding the presented use case with two buildings belonging to the same organisation, one can definitely associate "single authority" with the organisation, and "contributors" with the scientists who are working on the project and have admin access to the data. It is then more difficult with the "consumers" of the data. In our case the consumers are people who occupy the rooms, where the sensors and actuators are installed. We can not currently give access to the data to the "consumers", because we would provide all consumers with access to all data.

The access to the data via web server with secure hypertext transfer protocol (https) encryption and basic authentication is for example used by internet banking. However, two step authentication is used there nowadays. This could also be the next step to increase data security in the IoT field. As in the case of internet banking, it would significantly complicate the API and it would require a second channel for authentication. Another example of secure communication is implemented in the Smart Meter Gateways, where the communication between the "Things" and servers takes place over "Gateways". The security of the communication between the thing (normally counter) and the gateway depends on the communication type (eg WM-Bus). The communication between the gateway and servers is supposed to be very secure.

4. Evaluation

In this section the evaluation of usability of the software architecture shown in Figure 3 is presented in the light of two main application areas.

4.1 Centralized monitoring and control

The automatic operation of sensor data acquisition and control of available actuators was implemented as shown in Figure 3. There are around 100 of multisensors installed in two buildings, with several hundreds datastreams and 16 currently working actuators (MClimate Thermostat valves, Vicki). The planned software "digital janitor" is intended as a self-learning system which takes into account the relevant sensor values in order to produce optimized control commands for actuators, eg the heating system. As a first prototype to test the operation via open standards, a rule based program is implemented which controls the thermostat valves in building 2 in order to reduce the room temperature at night and on weekends.

At 7 pm every working day it copies the user-defined target temperature of each thermostat valve and sends the command to lower the target temperature to 16°C to all devices. At 6 am every working day the command "set the target temperature to the value which was stored" is sent. Such commands indeed consist of many HTTP requests. For example, the request to set the target_temperature of the thermostat valve with ID=9 to 20°C is shown in figure 5. The parameters are specified in JSON format {"taskingParameters": {"target_temperature": 20}} and sent as POST request to the specified URL and port of the web service. The exact table to be processed is specified as suffix "/TaskingCapabilities(9)/Tasks" to the URL. The web service receives the HTTP request and submits the changes to the DBMS via SQL and publishes the according message on the MQTT server as well. Our MQTT client "TTNtoFrost" receives the message and submits the command to the TTN server via MQTT to initiate the action of the corresponding LoRaWAN device, registered on TTN.

The operation of the valve and the dynamic of the room temperature are shown in Figure 6. The rooms are cooled down at night and during weekends and are heated only during working days. It turns out that due to thermal inertia the indoor temperature does not reach the target temperature on the first days after a weekend, but only on Wednesday, Thursday and Friday.

4.2 Visualisation with Grafana.

For simple monitoring purposes of sensor data and actuator status information Grafana is often used. In the data infrastructure presented here, Grafana provides access to live measurement data in the form of charts and dashboards (see for example Fig. 7) over the browser for many different users.

Grafana uses the "Frost Sensor Things API Plugin", which allows Grafana to read the json output format of the FROST server. However, the plugin couldn't reliably be used to retrieve metadata from Datastreams, Things and Locations tables as variables within Grafana. Instead, the generic plugin "JSON API" was successfully applied to extract the meta data for dashboards from the FROST server. Additionally, the plugin "PostgreSQL" allows much faster data transport from FROST to Grafana, but only in the local network, whereas the access over "Frost Sensor Things API Plugin" works from everywhere on the internet.

Grafana allows several methods of work with user accounts: 1) users can register on the Grafana server themselves, 2) an admin can add users manually, and 3) single sign-on. We use method 2). Currently, users are the people working in offices of building 2. Each office is equipped with a set of sensors (air quality and motion detectors) as well as with actuators (thermostat valves). Each user must receive access to the data of their own office and not the the data of any offices. To implement it, a dedicated dashboard for each user was created and permissions to this dashboard only provided to the user.

5. Conclusions

A system was implemented using the open standard Sensorthings API which performs monitoring and control of various devices with different technologies and protocols installed in several buildings. It could be shown that in principle the performance monitoring and sending of control commands are working, thus providing a usable basis for further work towards the intended applications.

However, some aspects of the STA were found to be not optimal for the use case. Namely, a possibility for hierarchical location coding (city, building, room, ...) would be helpful for structuring the things distributed in the buildings. The naming

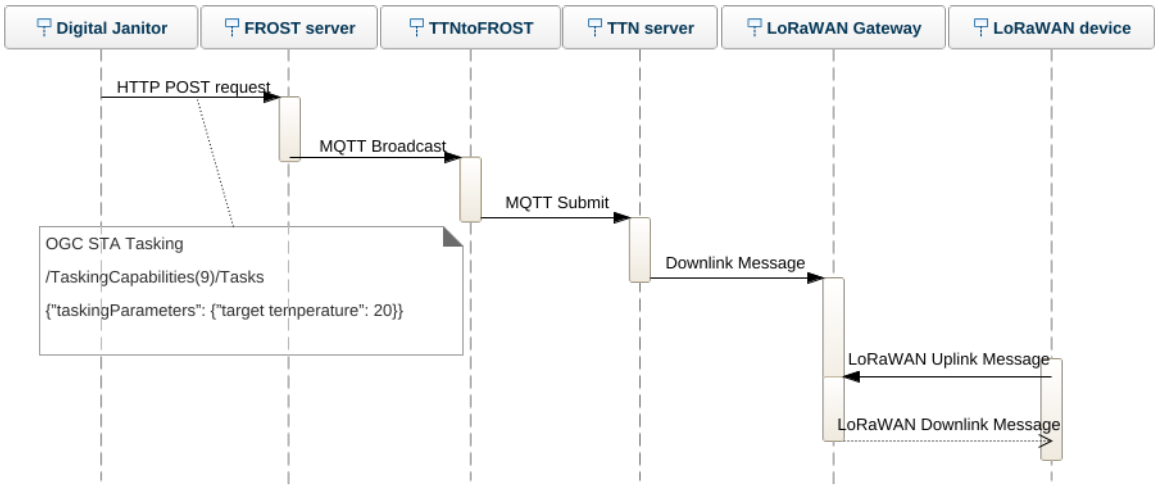

Figure 5. Example of actuator command (UML sequence diagram) for setting the target temperature of a LoRaWAN-equipped thermostat valve.

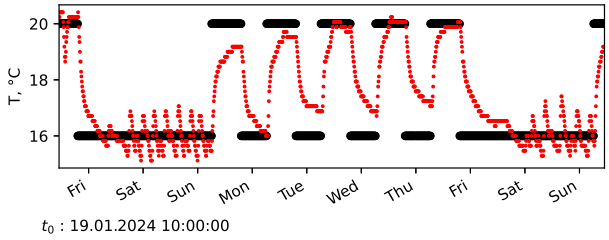

Figure 6. Demonstration of centralized monitoring and control over the system shown in Figure 3. Black - the dynamics of the target temperature of thermostat; red - the sensor temperature of thermostat in an office room of building 2.

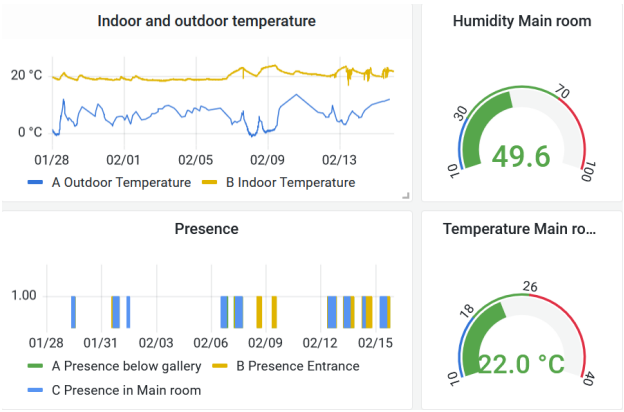

Figure 7. Example of a dashboard with different visualisations of sensor data from building 1 using Grafana. Shown are indoor and outdoor temperature time series as charts and presence as boolean values over time, and the actual values of indoor temperature and relative humidity as gauges.

convention implemented as a workaround revealed another opportunity for improvement: an option switch for ensuring the uniqueness of entity names.

With the decision to centralise the control system arises the problem of loyalty, as illustrated in chapter 2.6 of the book (Frochte, 2019). Applicable to our use case one can heat the buildings in two ways: 1 - to minimise the energy consumption of the local building, 2 - to minimise the overall energy consumption and to optimise the network load of the city. In the case 2 we will possibly sacrifice the interests of the local building in favour of common interests. Nevertheless, the control over the larger (than one building) scale may be reasonable, if all the bills for resources are paid centrally.

Acknowledgements

We are grateful to the German Federal Ministry of Education and Research for financial support via the project ”WärmewendeNordwest” (Förderkennzeichen 03SF0624). We are also grateful to our colleagues from the different partner organisations of the project for useful information exchange and discussions.

References

Atkinson, R., Zaborowski, P., Noardo, F., Simonis, I., 2022. Smart cities - systems of systems interoperability and OGC enablers. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, X-4/W3-2022, 19–26.

BSI Team, 2024. BSI TR-03109 Technische Vorgaben für intelligente Messsysteme und deren sicherer Betrieb. Federal Office for Information Security www.bsi.bund.de/EN/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/Technische-Richtlinien/TR-nach-Thema-sortiert/tr03109/TR-03109_node.html (visited 29.01.2024).

Chaturvedi, K., Kolbe, T. H., 2019. Towards Establishing Cross-Platform Interoperability for Sensors in Smart Cities. *Sensors*, 19(3). https://www.mdpi.com/1424-8220/19/3/562.

Elsys Team, 2020. Datasheet EMS Door. LoRaWAN Wireless Sensor. https://elsys.se/public/datasheets/EMS_Door_datasheet.pdf.

Emde, K., Budde, M., Fischer, T., Martin, T., Hilbring, D., 2022. Interaktive Steuerung der Ausführung von KI-Algorithmen in Umweltinformationssystemen über OGC SensorThings. INFORMATIK 2022.

European Union, 2024. General Data Protection Regulation. gdpr-info.eu (visited 29.01.2024).

Fang, H., 2015. Managing data lakes in Big Data era – what's a data lake and why has it became popular in data management ecosystem. *In: IEEE International Conference on Cyber Technology in Automation.*

Fischer, M., Gras, P., Löwa, S., Schuhart, S., 2021. Urban Data Platform Hamburg: Integration von Echtzeit IoT-Daten mittels SensorThings API. *ZFV (Zeitschrift für Geodäsie, Geoinformation und Landmanagement)*, 146(1), 47–56.

Fraunhofer IOSB Team, n.d. FROST®-Server - Open-Source-Implementierung der OGC SensorThings API. www.iosb.fraunhofer.de/de/projekte-produkte/frostserver.html (visited 24.01.2024).

Frochte, J., 2019. *Maschinelles Lernen. Grundlagen und Algorithmen in Python*. Carl Hanser Verlag München.

Gira Giersiepen GmbH &Co. KG, 2020. Gira IoT REST API Dokumentation. https://partner.gira.de/data3/Gira_IoT_REST_API_v2_DE.pdf (visited 04.02.2024).

Happ, M., Wielgosch, J., 2023. Open Data und Open Source für nachhaltige Smart-City-Lösungen. *Initiative Stadt.Land.Digital*. www.econstor.eu/handle/10419/276211.

Hevner, A., March, S., Park, J., Ram, S., 2004. Design Science in Information Systems. *MIS Quarterly*, 28, 75–105.

ISO Team, 2023. ISO/IEC 9075-1:2023 - Database languages SQL Part 1: Framework (SQL/Framework). International Standardization Organisation (ISO). www.iso.org/standard/76583.html (visited 24.01.2024).

ISO/IEC Team, 2006. ISO/IEC 14543-3-1:2006 - Information technology - Home electronic systems (HES) architecture. International Organization for Standardization, Geneva, Switzerland. https://www.iso.org/standard/43364.html.

Jarosch, H., 2016. *Grundkurs Datenbankentwurf Eine beispielorientierte Einführung für Studierende und Praktiker*. Springer Vieweg.

Krishnan, A., 2020. LoRaWAN and Multi-RAN Architecture Connecting the Next Billion IoT Devices. ABI Research, https://info.semtech.com/abi-research-white-paper.

Lindholm, O., Rehman, H. u., Reda, F., 2021. Positioning Positive Energy Districts in European Cities. *Buildings*, 11(1). https://www.mdpi.com/2075-5309/11/1/19.

LoRa Alliance, 2016. LoRaWAN. What is it? A technical overview of LoRa and LoRaWAN. Technical Marketing Workgroup 1.0. https://resources.lora-alliance.org/document/what-is-lorawan.

Mathis, C., 2017. Data Lakes. *Datenbank-Spektrum*, 17(3), 289–293.

MClimate Team, 2021. Datasheet Vicki by Mclimate. https://docs.mclimate.eu/mclimate-lorawan-devices/devices/mclimate-vicki-lorawan.

Muñiz, R., Del Coso, R., Nuño, F., Villegas, P. J., Álvarez, D., Martínez, J. A., 2024. Solar-Powered Smart Buildings: Integrated Energy Management Solution for IoT-Enabled Sustainability. *Electronics*, 13, 317.

OGC Team, 2019. OGC SensorThings API Part 2 – Tasking Core. Open Geospatial Consortium. docs.ogc.org/is/17-079r1/17-079r1.html.

OGC Team, 2021. OGC SensorThings API Part 1: Sensing Version 1.1. Open Geospatial Consortium. docs.ogc.org/is/18-088/18-088.html (visited 24.01.2024).

OGC Team, 2022. OGC Best Practice for using SensorThings API with Citizen Science. Open Geospatial Consortium. www.opengis.net/doc/bp/21-068 (visited 24.01.2024).

OGC Team, 2023. OGC SensorThings API 1.1 Extension:STAplus 1.0. Standard Implementation Draft. Open Geospatial Consortium. portal.ogc.org/files/103639 (visited 25.01.2024).

Orfanos, V. A., Kaminaris, S. D., Papageorgas, P., Piromalis, D., Kandris, D., 2023. A Comprehensive Review of IoT Networking Technologies for Smart Home Automation Applications. *Journal of Sensor and Actuator Networks*, 12(2). https://www.mdpi.com/2224-2708/12/2/30.

Sita, I.-V., Dobra, P., 2014. KNX Building Automations Interaction with City Resources Management System. *Procedia Technology*, 12, 212–219.

Sonnenberg, C., vom Brocke, J., 2012. Evaluation Patterns for Design Science Research Artefacts. *Proceedings of the European Design Science Symposium 2011*.

Team of EU parlament and council, 2023. Directive (EU) 2023/1791 of the european parlament and of the council of 13 September 2023 on energy efficiency and amending Regulation (EU) 2023/955 (recast). *Official Journal of the European Union*, L 231, L 231/1 – L 231/111.

Tee, B. T., Lim, S., Siew, P. W., Lee, M. F., 2023. Application of Sensor Technology for Energy Consumption Analysis: A Case Study in a Smart Office Building. *2023 IEEE International Conference 2023*, 0913–0917.

UDMS Team, 2024. Smart Data & Smart Cities 2024. https://udms.net/sdsc2024/ (visited 29.01.2024).

van der Schaaf, H., Moßgraber, J., Grellet, S., Beaufils, M., Schleidt, K., Usländer, T., Frysinger, S. P., Schimak, G., Knibbe, W. J., 2020. An Environmental Sensor Data Suite Using the OGC SensorThings API. I. N. Athanasiadis (ed.), *Environmental Software Systems. Data Science in Action*, Springer International Publishing, Cham, 228–241.

Wang, Z., Hong, T., 2020. Reinforcement learning for building controls: The opportunities and challenges. *Applied Energy*, 269, 115036.

Zhang, X., Shen, J., Saini, P. K., Lovati, M., Han, M., Huang, P., Huang, Z., 2021. Digital Twin for Accelerating Sustainability in Positive Energy District: A Review of Simulation Tools and Applications. *Frontiers in sustainable cities*, 3, 663269.