Chapter 11

Smart Building: The Tecnalia KUBIK Use Case

By Miguel Ángel Antón, Rubén Mulero, Sheila Puente, Larraitz Aranburu and Sarah Noyé

DOI: 10.1561/9781680838251.ch11

Published in *DevOps for Trustworthy Smart IoT Systems* by Nicolas Ferry, Hui Song, Andreas Metzger and Erkuden Rios (eds.). 2021. ISBN 978-1-68083-824-4. E-ISBN 978-1-68083-825-1.

Suggested citation: Miguel Ángel Antón, Rubén Mulero, Sheila Puente, Larraitz Aranburu and Sarah Noyé. 2021. “Smart Building: The Tecnalia KUBIK Use Case” in *DevOps for Trustworthy Smart IoT Systems*. Edited by Nicolas Ferry, Hui Song, Andreas Metzger and Erkuden Rios. pp. 241–258. Now Publishers. DOI: 10.1561/9781680838251.ch11.

now
the essence of knowledge

11.1 Introduction

Buildings have long been equipped with sensors and actuators to automate their control. Smart buildings are those whose facilities and systems (air conditioning, heating, lighting, access control systems, etc.) allow integrated and automated building management and control to increase energy efficiency, security, and usability. With the democratization of the Internet of Things (IoT), the number of sensors and actuators is constantly increasing, giving ways to new applications. The reduction of sensors and actuators cost is driving a digital shift in the building sector.

The need for better energy resource control and the requirement to provide better comfort for the user has led to a new market of complex Smart IoT Systems able to provide a vast array of new services or applications to the end-user. Extending legacy system to take advantage of those new services can be expensive, thus limiting possibilities. There is a need for a seamless way to integrate solutions from different manufacturers as well as to ensure effective design, deployment, and operation of simultaneous IoT applications that respect security and privacy requirements.

In that sense, software needs to be changed when new IoT devices are added or new functionalities for user comfort are developed. Therefore, it is necessary to ensure the continuous development and update of the IoT applications. The thermal and climate control needs to be continuously adapted to the environmental changes to keep the occupants' comfort in the buildings. At the same time, potential conflicts between IoT applications acting on the same actuator or the same physical parameters must be identified to guarantee the buildings' trustworthiness. And finally, cybersecurity threats need to be identified and mitigated to preserve the security and data privacy.

By leveraging the ENACT DevOps framework, secure and trustworthy IoT applications can be developed based on the interoperability and orchestrated operation of multiple sensors and actuators.

The smart building KUBIK, situated close to Bilbao, Spain, was inaugurated in 2010 as an experimental infrastructure for developing and validating innovative products and systems to optimize energy efficiency in buildings [1]. It is a three floors building owned by Tecnalia and designed for testing and research ranging from passive systems such as modular insulating components for roofs and facades, to energy generation based on renewable energy and climate control systems. It includes more than 700 sensors and actuators, central Building Management System (BMS), local Renewable Energy Systems RES (RES), local weather station, and Combined Heat and Power (CHP) equipment on-site. In the context of the ENACT project, KUBIK provides the required equipment and well-known boundary conditions for the testing and validation of the enablers developed in the project.

KUBIK experimental infrastructure is relevant to the ENACT project due to its special needs and characteristics such as the combination of legacy building automation systems and new smart IoT devices, this fact requires an interoperability platform to communicate both systems. At KUBIK, several energy efficiency applications and user comfort applications share common actuators (fancoils, lights, blinds, controlled sockets, etc.) which generate actuation control conflicts. Thermal control of a building is also a trade-off between energy consumption and user comfort that must be adjusted to the specific physical characteristics of the building and user preferences. Behavioral drifts in the control of building systems also need to be identified and addressed. And finally, security and privacy of the communications is a must, paying special attention in secure actuation.

The ENACT enablers in combination with the SMOOL middleware platform have been used to solve the challenges described in the previous paragraph. Now, IoT applications are designed, developed and improved using the DevOps strategy, as ENACT enablers ensure no actuation conflicts, security (secure communications, access control, threat detection, etc.) and trustworthiness (self-learning controls, behavioural drifts identification, etc.) saving time and effort.

As a general result of the ENACT project, KUBIK building was leverage to become a place to develop new applications for energy efficiency and user comfort for Smart Buildings that are based on IoT equipment. This exploitation of KUBIK as an infrastructure/testbed for the IoT domain is completely new. Now, the KUBIK building offers an environment where new IoT devices can be introduced and a wide range of IoT applications for smart buildings can be deployed and tested.

Section 11.2 describes the KUBIK building that was used as an experimental platform to validate the ENACT enablers for smart buildings. Section 11.3 gets into the detail of the technical architecture of the Smart IoT System of the KUBIK building. In Section 11.4, we expose different test scenarios and the benefit of the ENACT enablers. Finally, Section 11.5 concludes.

11.2 The KUBIK Smart Building

KUBIK is an experimental infrastructure focused on the development of new products and systems that provide energy consumption reduction for the building and increase user comfort (Figure 11.1). Its uniqueness lies in its ability to generate realistic scenarios to test energy efficiency resulting from the integration of constructive solutions, air conditioning and lighting systems, and energy supply from conventional and renewable energies. The building contains three floors with different testing zones and a cellar. Its ground floor is an apartment. It has a bedroom, a kitchen, a living room, and a corridor where engineers can test the Energy Efficient Building scenarios for a real home.

The ground floor has various IoT devices installed. Figure 11.2 shows the sensors and actuators installed at the ground floor. There is a flood sensor, sensors on doors

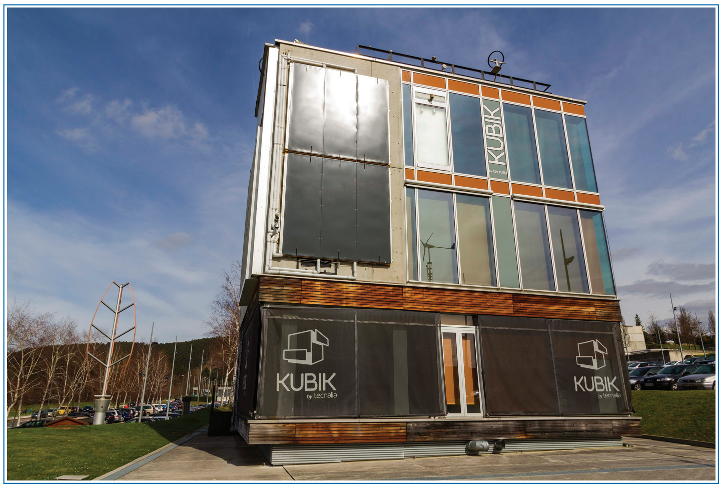


Figure 11.1. KUBIK by Tecnalia.

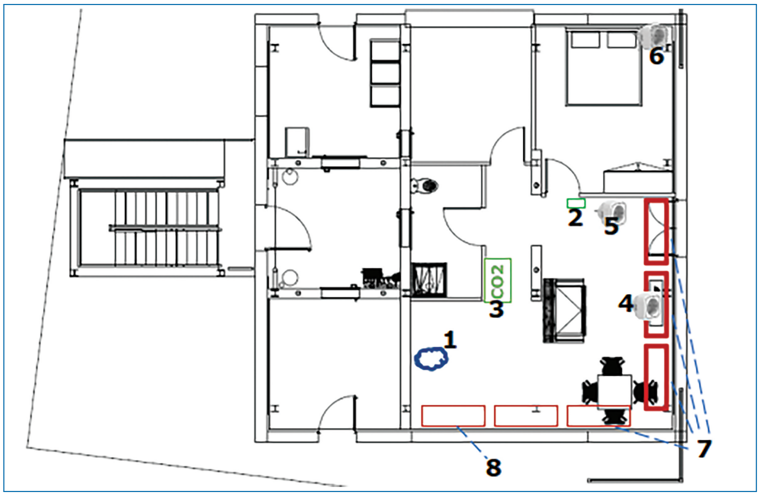


Figure 11.2. Floor plant of the Ground Floor of KUBIK.

Table 11.1. Device/Signals on the Ground Floor and Floor Plant of KUBIK Building. (K) values represents Kitchen, (B) values represents Bedroom, (L) value represents Living Room, (C) value represents corridor.

ID	Type	Device type	Location	System	Signal
1	Sensor	Water Flood	K	Z-Wave	Flood alarm state: ON/OFF
2	Sensor	Door Multisensor	B	Z-Wave	Position: OPEN/CLOSED
3	Sensor	CO2 Sensor 1	K/L/C	Z-Wave	CO2 level 0: 0 ppm to 200 ppm
4	Actuator	Remote Socket 1	L	Z-Wave	Switch state: ON/OFF
4	Sensor	Remote Socket 1	L	Z-Wave	Energy consumption: Watts
5	Actuator	Remote Socket 2	L	Z-Wave	Switch state: ON/OFF
5	Sensor	Remote Socket 2	L	Z-Wave	Energy consumption: Watts
6	Actuator	Remote Socket 3	B	Z-Wave	State: ON/OFF
6	Sensor	Remote Socket 3	L	Z-Wave	Energy consumption: Watts
7	Actuator	4 Blinds motors	L	PLC	Position: UP/Down
8	Actuator	2 Blinds motors	K	PLC	Position: UP/Down

that indicate open or closed status, various electrical sockets sensors and actuators, and motors for the blinds. Except for the blind motors, the devices are wireless sensors and actuators integrated as an additional layer to the building control system. Table 11.1 shows a detailed description of each device represented in the floor plan of the ground floor of KUBIK, its location in a specific room, its belonging to the IoT Smart Space or the wired Building Control group, and finally, the measures or commands it provides.

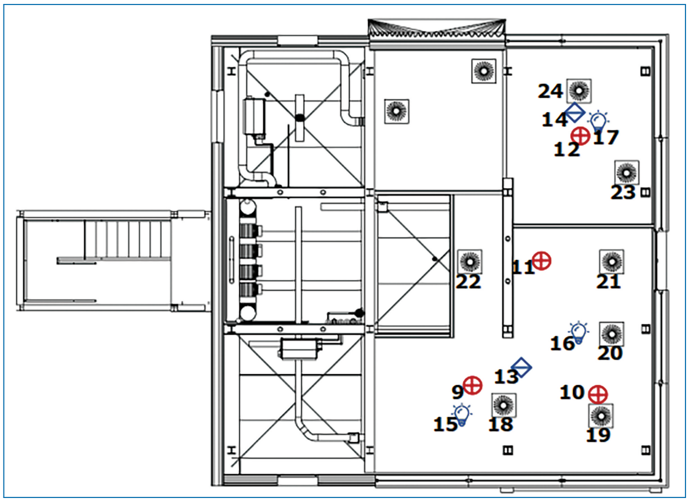


Figure 11.3. Reflected Ceiling plant of the Ground Floor of KUBIK.

Similarly, the reflected ceiling plan of the ground floor of KUBIK with sensors and actuators in their approximate location is shown in Figure 11.3. Those sensors are wireless Z-wave sensors monitoring ambient conditions (temperature, humidity, lighting, and occupancy) and smoke detectors. Sensors and actuators of the fan coil units of the space are connected to the building wired control system. Table 11.2 gives the details of different sensors and their signal types.

The cohabitation between hard-wired legacy sensors and easy-to-install additional wireless sensors presents an interesting case in line with the desire for flexibility and evolution of the smart building applications in the market.

11.3 Technical Architecture

A set of sensors, actuators, and devices are deployed inside each floor of the KUBIK building to capture real-time data and store it in a persistent environment. The stored data is analyzed to find potential solutions to automate different processes. Each connected device uses a standardized communication protocol to enable interoperability among them. A hub acts as a middleware between connected devices and external software programs and manages the communication protocol. The communication protocol may vary because of the installed devices' connectivity and cause additional complexity for large deployment scenarios. Therefore, it is necessary to create a general-purpose system to centralize the connections no matter what device type is connected.

Table 11.2. Devices/Signal on Ground Floor and Reflected Ceiling of KUBIK Building. (K) values represents Kitchen, (B) values represents Bedroom, (L) value represents Living Room, (C) value represents corridor.

ID	Type	Device type	Location	System	Signal
9	Sensor	Ceiling Multisensor 1	K	Z-Wave	Motion: YES/NO
9	Sensor	Ceiling Multisensor 1	K	Z-Wave	Temperature: degrees Celsius
9	Sensor	Ceiling Multisensor 1	K	Z-Wave	Light: 0 lux–1000 lux
9	Sensor	Ceiling Multisensor 1	K	Z-Wave	Relative humidity: 20%–95%
10	Sensor	Ceiling Multisensor 2	L	Z-Wave	Motion: YES/NO
10	Sensor	Ceiling Multisensor 2	L	Z-Wave	Temperature: degrees celsius
10	Sensor	Ceiling Multisensor 2	L	Z-Wave	Light: 0 lux – 1000 lux
10	Sensor	Ceiling Multisensor 2	L	Z-Wave	Relative humidity: 20%–95%
11	Sensor	Ceiling Multisensor 3	L/C	Z-Wave	Motion: YES/NO
11	Sensor	Ceiling Multisensor 3	L/C	Z-Wave	Temperature: degrees Celsius
11	Sensor	Ceiling Multisensor 3	L/C	Z-Wave	Light: 0 lux–1000 lux
11	Sensor	Ceiling Multisensor 3	L/C	Z-Wave	Relative humidity: 20%–95%
12	Sensor	Ceiling Multisensor 4	B	Z-Wave	Motion: YES/NO
12	Sensor	Ceiling Multisensor 4	B	Z-Wave	Temperature: degrees Celsius
12	Sensor	Ceiling Multisensor 4	B	Z-Wave	Light: 0 lux – 1000 lux
12	Sensor	Ceiling Multisensor 4	B	Z-Wave	Relative humidity: 20%–95%
13	Sensor	Smoke Detector 1	K/L/C	Z-Wave	Alarm state: ON/OFF
14	Sensor	Smoke Detector 2	B	Z-Wave	Alarm state: ON/OFF
15	Actuator	Ceiling light 1	K	PLC	Light state: ON/OFF
16	Actuator	Ceiling light 2	L	PLC	Light state: ON/OFF
17	Actuator	Ceiling light 3	B	PLC	Light state: ON/OFF
18	Actuator	Fan Coil 1	K	PLC	Operation state: ON/OFF
18	Actuator	Fan Coil 1	K	PLC	Temperature setpoint: Celsius
19	Actuator	Fan Coil 2	L	PLC	Operation state: ON/OFF
19	Actuator	Fan Coil 2	L	PLC	Temperature setpoint: Celsius
20	Actuator	Fan Coil 3	L	PLC	Operation state: ON/OFF
20	Actuator	Fan Coil 3	L	PLC	Temperature setpoint: Celsius
21	Actuator	Fan Coil 4	L	PLC	Operation state: ON/OFF
21	Actuator	Fan Coil 4	L	PLC	Temperature setpoint: Celsius
22	Actuator	Fan Coil 5	C	PLC	Operation state: ON/OFF
22	Actuator	Fan Coil 5	C	PLC	Temperature setpoint: Celsius
23	Actuator	Fan Coil 6	B	PLC	Operation state: ON/OFF
23	Actuator	Fan Coil 6	B	PLC	Temperature setpoint: Celsius
24	Actuator	Fan Coil 7	B	PLC	Operation state: ON/OFF
24	Actuator	Fan Coil 7	B	PLC	Temperature setpoint: Celsius

The devices installed in KUBIK use two different communication protocols. The first one is called Z-Wave,[1] a wireless communication protocol that integrates smart sensors inside a building. Z-Wave devices are widely used in domestic environments due to their ease of installation and low cost. However, its signal quality can be affected by interference and its battery level. The second one is called the MODBUS[2] communication protocol. This protocol is an industry-standard that is robust, fast, and secure. The connected devices using the MODBUS communication protocol need a central node called *industrial PC* or Programmable Logic controller.[3] They need to be connected by a physical connection (a wired cable). The MODBUS communication protocol is widely used in industrial processes to obtain information from machines and active processes. Thus, the main difference between a Z-Wave and a PLC device is that the former does not need any physical connection, and the latter requires a physical connection to an industrial PC. In terms of installation, Z-Wave-based devices are more convenient than a PLC device, but a PLC device offers robust connectivity and high reliable data speeds.

Having two different communication protocols to acquire data or perform actuation processes, we need to implement a middleware that enables interoperability among various sensors and actuators. The SMOOL IoT middleware that has a semantic broker for connecting heterogeneous devices or sources of information. In addition, the Building Management System also centralize all the information of new wireless sensors/actuators and the legacy building control systems of the building using a Scada software.

To address the challenges of sharing sensors/actuators between IoT applications that are running at the same time and also add a security layer to the data streams, the ENACT project provides tools and enablers to ensure the trustworthiness of the IoT applications in the KUBIK infrastructure. Figure 11.4 depicts the high-level architecture of the communications architecture in the KUBIK building.

The high-level architecture of the communications in KUBIK building is divided into three modules. The first module contains two subsystems: (i) **system 1**, the *wireless* system where each device is connected to a central node or *network hub* managing wireless connections; and (ii) **system 2**, a *wired* system where each device is physically connected to a central node or PLC device managing each wired connection. The second module contains the Building Management System (BMS) having three main elements: (i) a gateway device managing the connections between PLC/Z-Wave nodes, (ii) a persistent database that gathers the

[1] https://www.z-wave.com/

[2] https://modbus.org/

[3] https://en.wikipedia.org/wiki/Programmable_logic_controller

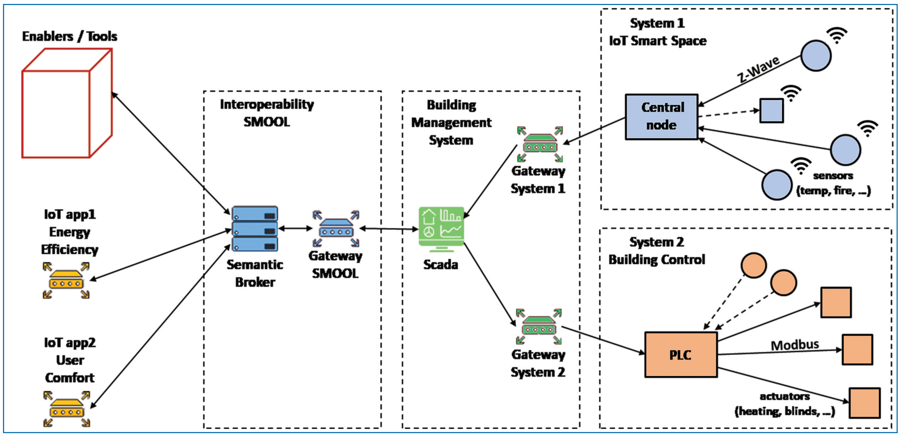


Figure 11.4. High level communications architecture in the KUBIK building.

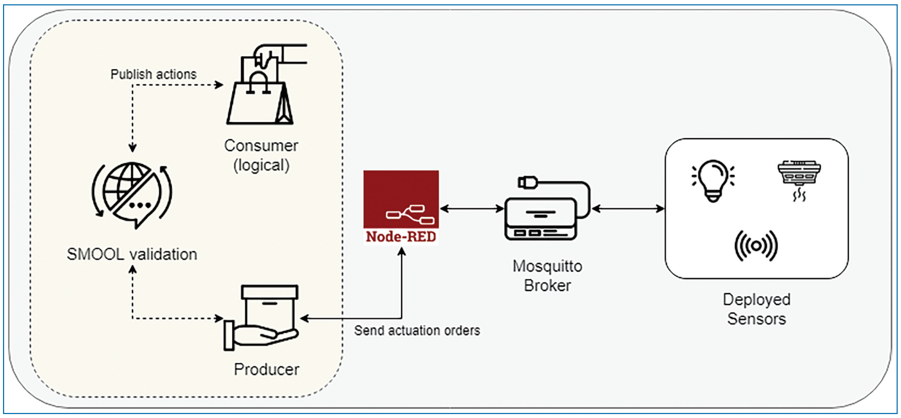
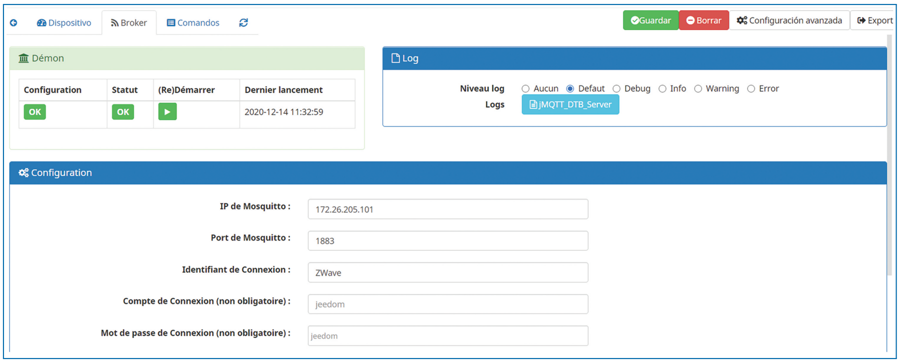


Figure 11.5. Technical components used in KUBIK architecture.

total connected devices and stores the data read by each device, (iii) a SCADA interface that graphically shows the current status of each deployed device and enables engineers to update device configurations. The last module is the SMOOL interoperabilty layer [2]. The entire system is managed by the decisions made according to a set of rules programmed in the IoT applications.

Figure 11.5 depicts the technical components used in the above architecture. The "Deployed sensors" part represents the acquisition process in which a set of sensors using both Z-Wave (system 1) and legacy PLC (system 2) obtain data measures to be sent to a middle device called "Mosquitto MQTT broker".[4]

[4] https://mosquitto.org/



Configuration	Statut	(Re)Démarrer	Dernier lancement
OK	OK	▶	2020-12-14 11:32:59

Niveau log ○ Aucun ● Defaut ○ Debug ○ Info ○ Warning ○ Error
Logs jMQTT_DTB_Server

IP de Mosquitto : 172.26.205.101
Port de Mosquitto : 1883
Identifiant de Connexion : ZWave
Compte de Connexion (non obligatoire) : jeedom
Mot de passe de Connexion (non obligatoire) : jeedom

Figure 11.6. Jeedom configuration interface.

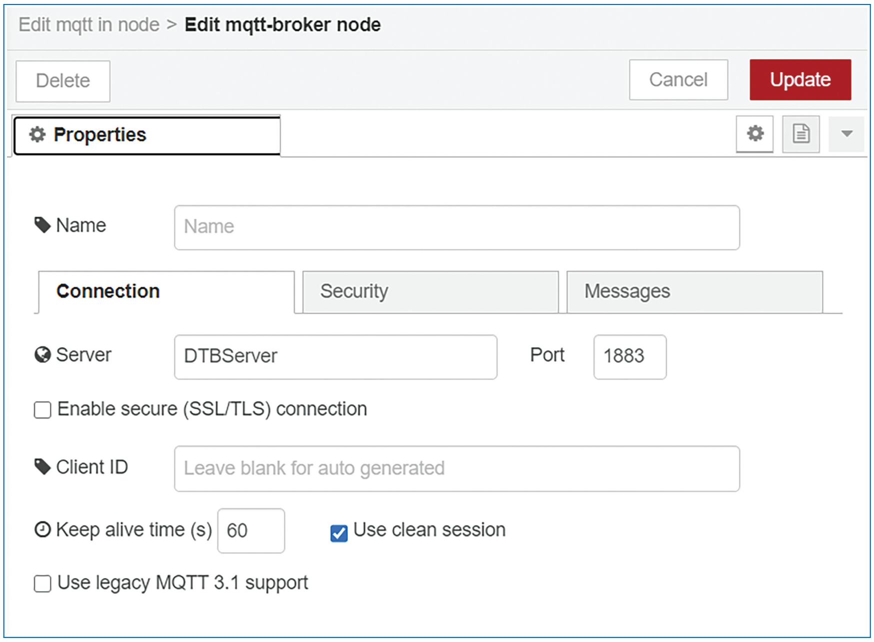
The SMOOL middleware has two components, i.e., the Producer and Consumer modules. The Producer module sends secure actuations when the Consumer module requires a security token. The Consumer module executes a set of expert rules using the obtained data from the Producer module. An expert encodes the expert rules. He/she decides which recommended actions to take when the data acquired from sensors meets a condition. For example, if the illumination sensors detect too much light, the actuators open the blinds for sunlight to get inside the KUBIK's living room.

At the low-level in the smart building Architecture (Figure 11.5), the data acquisition and actuation processes are managed by some hub systems called Gateways. These hubs are configured to allow direct communication between different sensors/actuators and a Mosquitto MQTT broker. Each hub uses its communication protocol to acquire or send actuation orders to the target device. For example, one hub is configured to manage only Z-Wave communications towards connected devices, while another hub is configured with the MODBUS communication standard. These hub systems use an internal operating system, JEEDOM,[5] that enables a graphical configuration of the connected devices and external services. In this regard, each hub is configured to make a direct connection to the Mosquitto MQTT broker. Figure 11.6 shows how JEEDOM is configured to send the data read from a set of Z-Wave devices directly to the Mosquitto MQTT broker.

Once the connection between the Z-Wave/PLC hub (JEEDOM) and Mosquitto broker is established, the next step is to program the Node-RED programming tool to develop a bridge between MQTT messages, SMOOL middleware and logic of IoT applications. Node-RED[6] is a visual flow-based programming environment

[5] https://www.jeedom.com/site/en/index.html

[6] https://nodered.org/



Name: Name

Server: DTBServer Port: 1883

□ Enable secure (SSL/TLS) connection

Client ID: Leave blank for auto generated

Keep alive time (s): 60 ☑ Use clean session

□ Use legacy MQTT 3.1 support

Figure 11.7. Node-RED programming to connect to Mosquitto MQTT Broker.

designed by IBM for the Internet of Things. Figure 11.7 depicts the configuration parameters needed to connect the Node-RED with the Mosquitto broker. These parameters enable the configuration of different workflows to make the required subscriptions to each connected device.

After having the logical connection between Node-RED and Mosquitto, it is necessary to program the required Node-RED flows for the data acquisition process with the SMOOL module. Figure 11.8 shows the acquisition process flow programmed in Node-RED. It reads the data from the Mosquitto broker and exposes it directly to its internal REST API module. The Producer SMOOL component reads the data exposed in the Rest API endpoints. Each connected line in Figure 11.8 represents a device inside the KUBIK building. There are several sensors deployed in the KUBIK building, and each one has its action flow. For convenience, Figure 11.8 presents only a minor part of the devices.

Figure 11.9 exhibits the secure actuation process provided by SMOOL middleware having two action flows: (i) one flow to send the available orders from Node-RED to SMOOL using a security token (top of the image) and (ii) another flow to sent the secure actuation orders checked in SMOOL to the actuator via Mosquito broker (low part of the image).

The flows are configured to enable the SMOOL components to send actions to the hub and write the needed information to perform the move up or move down actuation orders of the living room/kitchen blinds.

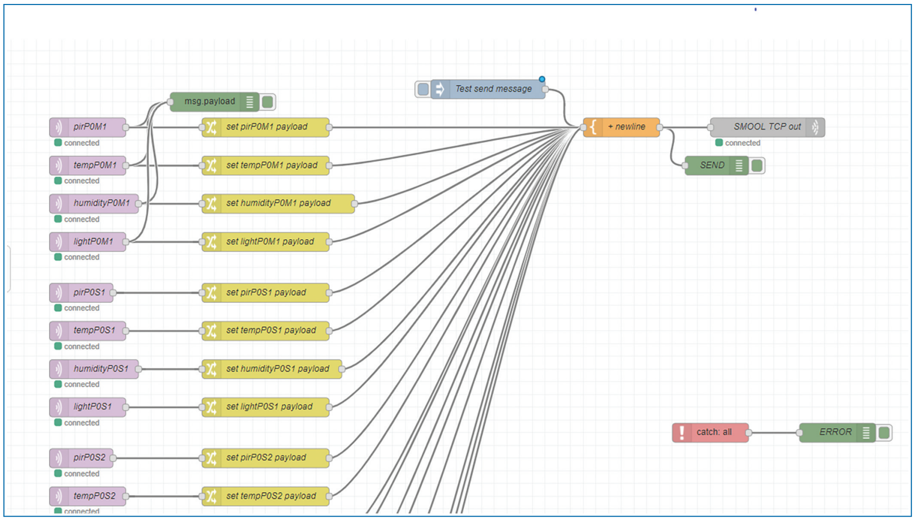


Figure 11.8. Node-RED flows to publish sensorized data in SMOOL.

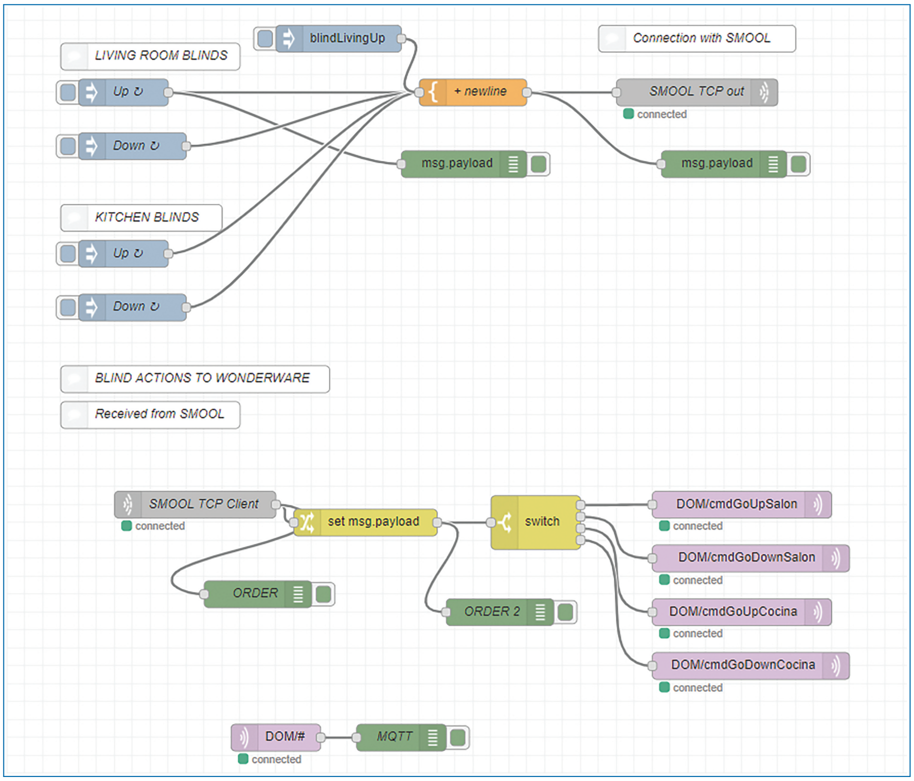


Figure 11.9. Node-RED flows to execute orders over devices in KUBIK through SMOOL.

11.4 KUBIK as an Experimental Platform for the ENACT Scenarios

In the ENACT project, the KUBIK building has been used as an experimental platform to test enablers in the *Smart Building* use case. Multiple scenarios have been implemented to test the ENACT tools relevant to this use case. We give the details of the scenarios and their main results in the following section. Those scenarios involve concurrent IoT applications related to energy efficiency and user comfort. These applications implement thermal comfort controls and smart building alerts that assure the safety of their occupants. They employ Z-wave and PLC systems described in the previous section.

11.4.1 Scenario 1: Thermal Comfort Control – Heating Design

Thermal comfort control is an essential smart building function. Sensors measure the users' comfort and enable the HVAC control system to keep the temperature at the level requested. By adding thermostat and temperature sensors with another protocol than the one of the HVAC system, it is easier to retrofit old systems and give more flexibility to deploy sensors. On the other hand, this strategy poses potential threats and risks to the thermal comfort system that must be identified and addressed.

One of the threats identified is when one of the thermal control devices is replaced with a similar device but not the same one. The Risk Management enabler is then used to analyse potential threats to the HVAC control system when new IoT devices are combined with legacy systems, provide the list of mitigation actions that the new IoT device needs to fulfil, and support the selection of security controls to minimize risks. By means of this enabler, HVAC control system designers and maintainers can also decide the risk level that that it is tolerable.

The retrofitting of old HVAC control systems is usually combined with changing the logic of the HVAC control program. In addition, the SMOOL middleware is used to communicate with new IoT devices and ThingML language can program those devices to define system behaviours and generation of executable code. The Orchestration and Continuous Deployment enabler, aka. GeneSIS, enables the continuous deployment and update of applications. In the ENACT project, GeneSIS has been fully integrated with SMOOL semantic middleware and ThingML language. In that way, SMOOL and ThingML are automatically deployed by GeneSIS as any other software component when adding new IoT devices and changing the HVAC control logic. The programming of the HVAC control logic, the communication characteristics of the new IoT devices in SMOOL and the programming of ThingML devices are done as part of the same project in the same Integrated Development Environment (IDE).[7]

The Orchestration and Continuous Deployment enabler has also been used with the Actuation Conflict Manager enabler in Scenario 2.

11.4.2 Scenario 2: Thermal Comfort Control – Conflict in Heating Actuator Use

In the continuation of scenario 1, the integration of multiple systems can result in two or more applications sending different temperature preferences to the same heating actuator, which causes a fluctuating operation of the thermal control. The Actuation Conflict Management (ACM) enabler allows findings at design time the actuation conflicts that may lead to these fluctuations, and then it helps solving the conflict, thereby fixing the cause of the fluctuation.

A direct conflict occurs when two applications try to access the same node, e.g. an actuator. These two applications accessing the same actuator might send contradictory commands resulting in an indeterministic behaviour. An example of direct actuation conflict has been programmed and tested in the ACM enabler (see Figure 11.10). Figure 11.10 shows two IoT applications fed with temperature values from two different sensors. The applications try to change the actuator state when the temperature values reach a threshold that mimics the thermostat operation. The temperature sensors are deployed at different parts of the room, and thus different temperature values are likely obtained. The actuator behavior is similar to a relay that can switch ON and OFF the thermal heater.

Figure 11.11 shows a Node-RED flow with the implementation of the previously described scenario. In that flow, two identical subflows represent sensor processes that access the "command" node to send an actuation command to the same heating

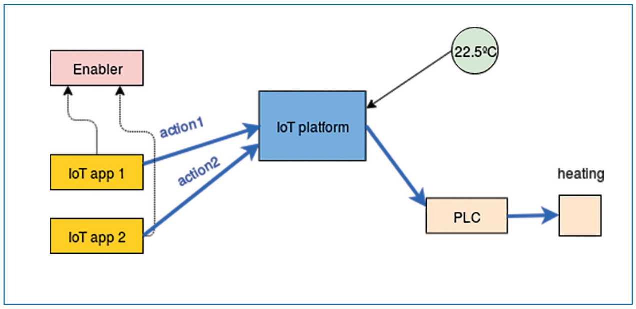


Figure 11.10. Thermal comfort control with conflict in heating actuator use.

[7] Details about this integration can be seen in the following video https://www.youtube.com/watch?v=mfT_AwfkXNc

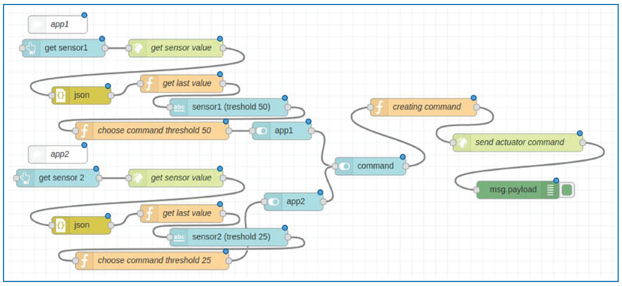


Figure 11.11. Node-RED flow example for thermal comfort control to be used by the ACM enabler.

actuator. The ACM enabler helps the software developer solve actuation conflicts occurring when two or more concurrent IoT applications try to send simultaneously conflicting actuation orders to the same actuator. In this scenario, a docker component that contains the Node-RED flows of the IoT application is deployed using GeneSIS. The ACM enabler then imports the model of the IoT system created by GeneSIS. GeneSIS enables the creation of an architecture and deployment model of the Smart IoT System by adding components and links. Although several model formats can be imported into the ACM enabler, Node-RED and GeneSIS are the main tools supported by the ACM enabler (Figure 11.11).

Once everything is imported and set up in the ACM enabler, a click on the "find conflicts" button automatically detects actuation conflicts and add a placeholder for actuation conflict management component in the model where conflicts might happen. The ACM enabler proposes several out-of-the-box components to solve a detected actuation conflict. The user chooses the right component, and the actuation conflict management placeholder is automatically replaced accordingly. Then, the IoT application is updated to become a conflict-free thermal comfort control system that can be redeployed using GeneSIS.

11.4.3 Scenario 3: Luminosity Comfort Control – Indirect Conflict in Luminosity Level Actuation

Two or more applications may also send actuation orders to different actuators that cause an indirect actuation conflict. An indirect actuation conflict affects the building thermal control of a building when two IoT applications managing two actuators act concurrently over the same physical variable, e.g. setting a high-temperature setpoint to the HVAC and opening a window. For instance, one app opens the window (lower temperature), and another one increases the setpoint temperature in the HVAC control system (higher temperature). An indirect actuation conflict can also occur in the room's luminosity level control when acting on lights and blinds. One switch controls the light to set it set ON or OFF, and another one controls the blind

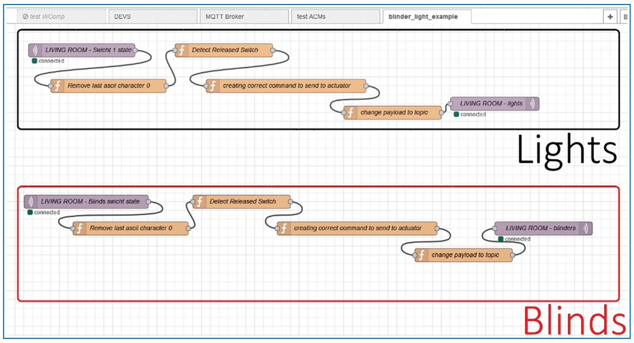


Figure 11.12. Node-RED flow representing the classic behaviour of the control of lights and blinds using switches.

to set it UP or DOWN. A physical model of the building is needed to find the indirect conflict on brightness, i.e. the impact of lights and blinds on the luminosity level.

The Actuation Conflict Management (ACM) enabler also corrects the operation of a physical system subject to uncertainties to deal with indirect conflicts. To achieve that, the physical system model is added to the enabler to detect and solve such indirect actuation conflicts. The ACM enabler imports the IoT system model created using GeneSIS and Node-RED tools. Once the model is imported, a physical process configuration is specified. The configuration allows establishing the interaction between a logical node and the environment. For instance, when there is an activation of a light, it is linked to a physical process representing luminosity. The utility of this process is to find an indirect conflict. If two different actions are linked to the same physical process, it may be an unplanned conflict to be solved.

In this scenario, the control of lights and blinds are associated with their corresponding switches using Node-RED. The classic behaviour of the system is that the lights are controlled by one switch to set it ON or OFF and the blinds are controlled by another switch to set it UP or DOWN (see Figure 11.12).

The ACM enabler then is fed with the previous described Node-RED flow for classic behaviour and the physical representation of the system to find actuation conflicts, i.e. the impact of lights and blinds on the luminosity level. Then, the ACM tool detects an actuation conflict in the luminosity level when we turn on the lights having sufficient brightness outside. The ACM tool creates a new Node-RED flow that resolves that conflict by adding new ACM components between the application logic and the actuation command (see Figure 11.13). The modified new Node-RED flow for luminosity level control can then be redeployed. The updated behaviour open the blinds instead of turning on the lights when the outside luminosity is high enough. Also in this sceanario, the Behavioural Drift Analysis

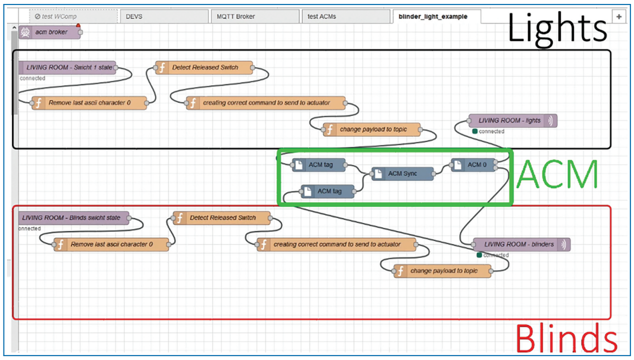


Figure 11.13. Node-RED flow example for solving the indirect conflict in luminosity level actuation.

(BDA) enabler was used to evaluate how much the observed behaviours of the IoT System are different from the expected ones.

11.4.4 Scenario 4: Smart Building Alerts for User Comfort

Data integrity and confidentiality must be ensured during data communication, especially if the communication is an important alert about equipment and user safety. Therefore, tampering with the alarms by external people should be avoided. The Security & Privacy Monitoring & Control (S&P Mon&Con enabler, cf. Chapter 7) enabler ensures the integrity and privacy of the communications through two services: (i) the Security and Privacy Monitoring for the surveillance of data security and (ii) the Security and Privacy Adaptation for data security enforcement.

This scenario addresses smart building alerts for IoT apps monitoring building aspects, such as an abnormally high or low temperature and smoke presence. The Security and Privacy Monitoring and Control (S&P Mon&Con) enabler ensures that the alarm system is not tampered with by external people and safeguard people's privacy. These enablers have been applied to the communication architecture in Figure 11.4.

The Security & Privacy Monitoring & Control enabler uses port mirroring to send a copy of the network packets that contain sensitive information of the smart building to a server where the enabler is running. Three different nodes in the smart building communication architecture have been mirrored: (i) the SMOOL communication broker, (ii) the Raspberry Pi used as a gateway by the SMOOL, and (iii) the SCADA node of the Building Management System of the KUBIK building.

When sensing information is transmitted, the S&P Mon&Con enabler checks if the messages are sent by the authorized nodes that belong to the KUBIK network. If an external device tries to send or receive a message to/from the monitored nodes, the S&P Mon&Con enabler notifies the system administrator to block network traffic if necessary. The S&P Mon&Con enabler also ensures the sensing data integrity by blocking any attempt of tampering with the message.

Scenario 4 addresses sensor devices that trigger alarms. These alarms may also trigger a security action or siren. The security actions consist of an ON/OFF command to a siren.

11.4.5 Scenario 5: Thermal Comfort Control – Self-optimizing Controller Design

The thermal comfort control of a building is a trade-off between user comfort and energy consumption, e.g., the heating is off when the room or building is not occupied. Monitoring the building's thermal inertia and real-time occupation can significantly improve energy efficiency and comfort compared to traditional controllers. We used the Online Learning enabler (OLE) to find and update the optimal control parameters at run time. We employed GeneSIS to properly deploy those control parameters and orchestrate the involved IoT devices.

The Online Learning (OLE) enabler is a module with an agent-based Artificial Intelligence (AI) algorithm that performs actions based on the sensors readings gathered from the KUBIK building infrastructure. This enabler aims to provide a thermal comfort solution that reduces the energy costs in the building. We have developed a thermal model of the ground floor of the KUBIK building to implement the OLE. We have demonstrated the potential for energy saving and increased thermal comfort.

11.5 Conclusion

Buildings are becoming increasingly smart, and new IoT tools are needed to ensure the safe and trustworthy operation of the different services they offer. The KUBIK experimental building is a smart building testbed for the ENACT project tools.

We successfully tested the ENACT enablers that enable continuous deployment, solving actuation conflicts, ensure security and privacy of the communications, identify risks at design time, correct behavioral drifts, enforce security and privacy, and self-optimizing controller.

References

[1] José A. Chica *et al.* “Kubik: Open Building Approach for the Construction of an Unique Experimental Facility Aimed to Improve Energy Efficiency in Buildings”. In: *Open House International* 36.1 (Jan. 2011), pp. 63–72. ISSN: 0168-2601. DOI: 10.1108/OHI-01-2011-B0008.

[2] Adrian Noguero, Angel Rego, and Stefan Schuster. “Towards a smart applications development framework”. In: *Social Media and Publicity* 27 (2014), pp. 2011–2020.