D^2EPC IoT Platform & Interfaces v1

The D^2EPC project has received funding from the EU's Horizon 2020 research and innovation programme under grant agreement No 892984

Project Acronym:	D^2EPC
Project Full Title:	Next-generation Dynamic Digital EPCs for Enhanced Quality and User Awareness
Grant Agreement:	892984
Project Duration:	36 months (01/09/2020 – 31/08/2023)

DELIVERABLE D3.1
D^2EPC IoT platform & Interfaces v1

Work Package:	WP3 - Building digitalisation and inverse modelling for implementing next generation dEPCs
Task:	T3.1 - IoT and BMS interfaces to extract energy related data
Document Status:	Final
File Name:	D^2EPC_D3.1_IoT Platform and Interfaces_v1
Due Date:	30/06/2021
Submission Date:	07/07/2021
Lead Beneficiary:	HYP

Dissemination Level	
Public	☒
Confidential, only for members of the Consortium (including the Commission Services)	☐

Authors List

Leading Author				
First Name		Last Name	Beneficiary	Contact e-mail
Thanos		Kalamaris	HYP	t.kalamaris@hypertech.gr
Co-Author(s)				
#	First Name	Last Name	Beneficiary	Contact e-mail
1	Thomas	Papapolyzos	HYP	thomas@hypertech.gr
2	Apostolos	Tsolakis	CERTH	tsolakis@iti.gr
3	Dimosthenis	Ioannidis	CERTH	djoannid@iti.gr
4	Panagiota	Chatzipanagiotidou	CERTH	phatzip@iti.gr
5	Stavros	Koltsios	CERTH	skoltsios@iti.gr
6	Gerfried	Cebrant	SEC	Gerfried.Cebrat@senercon.de
7	Mija	Susnik	DMO	Mija@demobv.nl
8	Christiana	Panteli	CLEO	cpanteli@cleopa.de
9	Karen	Semmler	CLEO	ksemmler@cleopa.de
10	Detlef	Olschewski	CLEO	dolschewski@cleopa.de
11	Phoebe	Georgali	FRC	res.gp@frederick.ac.cy
12	Paris	Fokaides	FRC	eng.fp@frederick.ac.cy

Reviewers List

Reviewers			
First Name	Last Name	Beneficiary	Contact e-mail
Christiana	Panteli	CLEO	c.panteli@cleopa.de
Phoebe Zoe	Morsink-Georgali	FRC	res.gp@frederick.ac.cy

Version History

v	Author	Date	Brief Description
0.1	Thanos Kalamaris, HYP	May 04, 2021	Initial draft (ToC)
0.2	Thanos Kalamaris, HYP	May 31, 2021	Partner's Contributions collection
0.3	Thanos Kalamaris, HYP	June 10, 2021	Draft version with initial content for all sections
0.5	Thanos Kalamaris, HYP	June 20, 2021	First draft ready for review
0.7	Dimosthenis Ioannidis, CERTH	June 30, 2021	Quality check and minor corrections
1	Thanos Kalamaris, HYP	July 5, 2021	Final version including comments from partners ready for submission

Legal Disclaimer

The D^2EPC project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 892984. The sole responsibility for the content of this publication lies with the authors. It does not necessarily reflect the opinion of the Executive Agency for Small and Medium sized Enterprises (EASME) or the European Commission (EC). EASME or the EC are not responsible for any use that may be made of the information contained therein.

Copyright

Executive Summary

The objective of D3.1 is to present the overall methodology of T3.1 and summarize the activities conducted until M10 towards the establishment of the D^2EPC IoT Platform. The aim of T3.1 "*IoT and BMS interfaces to extract energy-related data"* is to deliver the entire IoT framework planned to enable the continuous, seamless, and non-intrusive collection of energy consumption and ambient conditions data from the D^2EPC demonstration buildings, necessary to fulfill the project's objectives.

Within D3.1, the D^2EPC IoT framework is described in detail, expanding the description provided in the D^2EPC architecture [1], as defined in T1.4 of WP1. According to this description, the IoT framework was separated into conceptual layers describing both its hardware and software infrastructure. The physical layer contains the smart monitoring system comprising a set of IoT sensing devices and building management systems that communicate through wireless communication with the components in the interoperability layer. The interoperability layer consists of the Information Management Layer (IML) component responsible for collecting, processing, and streaming the data from the physical layer to the project's repository, where they are stored and distributed to other D^2EPC components. The IoT Gateway is the local hardware component acting as a bridge between the physical and the interoperability layer. The Gateway is capable of integrating a plethora of multiprotocol IoT devices and providing a common interface to the IML.

Within T3.1, an optimized IoT topology per pilot building will be proposed taking into account the data requirements of the project and the infrastructure availability on site. Based on this topology, the finalized bills of materials per site with off-the-shelf IoT devices will be defined within WP5 activities. The high-level methodology that defines the per-pilot IoT infrastructure can be described as a step-by-step approach based on the requirements arising from the key performance indicators and the common information model (elicited within WP2), along with the information extracted from the pilot audits in WP5. More specifically:

- Identification of the metrics & respective measurement granularities necessary for the KPI calculation, which will shape the requirements in terms of IoT infrastructure per pilot building
- Detailed and in-depth analysis of the audits on the pilot sites (performed under WP5), which will allow a complete picture of the available devices, the limitations per site, the areas to be included in the D^2EPC activities etc.; Through this information, the individual needs in IoT equipment per pilot site will be highlighted.
- Based on the above, extensive testing will be conducted to ensure both the effectiveness and applicability of the proposed off-the-shelf products and the interfacing with the available infrastructure whenever possible.

Initially, the D3.1 provides a summary of the pilot Infrastructure. Due to limited information availability on the submission month (M10), the current version includes only a high-level description of the infrastructure available at the pilot sites, including their short description, assets, and existing metering equipment with the respective information retrieval method. The deliverable further dedicates a chapter to the components of the IoT Equipment Framework and the respective equipment evaluation criteria set to overcome the obstacles that occur during the definition and deployment of an IoT Network. The final chapter presents the Information Management Layer component along with its technological capabilities that address the challenges related to big data directly linked to the IoT domain. Lastly, the IML and the IoT Equipment Framework are defined within the D^2EPC System Architecture.

The second and final version of the deliverable (D3.4, to be submitted in M16) will include information on the overall dynamic metric and system operation requirements defined by WP2 and a consolidated version with the IoT infrastructure requirements of the demonstration buildings. Furthermore, D3.4 will document the final specifications and topology of the distributed IoT components to be used, as

well as the description and outcomes of the lab or/and remote trials of the proposed IoT equipment. Finally, the second version of this document will describe the incorporation of the Information Management Layer in the D^2EPC data model.

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

Term	Description
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
BACnet	Building Automation and Control Networking Protocol
BIM	Building Information Modelling
BLE	Bluetooth Low Energy
BMS	Building Management System
BoM	Bill of Materials
CAN	Controller Area Network
DBSCAN	Density-based spatial clustering of applications with noise
DHW	Domestic Hot Water
EPC	Energy Performance Certificate
ESS	Energy Storage System
HDMI	High-Definition Multimedia Interface
HVAC	Heating, ventilation, and air conditioning
IML	Information Management Layer
IoT	Internet of Things
JSON	JavaScript Object Notation
kWh	Kilo Watt Hours
Li-ON	Lithium-Ion
LoRa	Long Range
MAD	Median Absolute Deviation
MQTT	Message Queuing Telemetry Transport
MWFA	Moving Window Filtering Algorithm
NBIoT	Narrow Band-Internet of Things
OEM	Original Equipment Manufacturer
PV	Photovoltaic
RTU	Remote Terminal Unit

REST	**Representational State Transfer**
RMS	**Root Mean Square**
RPC	**Remote Procedure Call**
R&D	**Research & Development**
SCADA	**Supervisory Control and Data Acquisition**
TCP	**Transmission Control Protocol**
UML	**Unified Modelling Language**
USB	**Universal Serial Bus**
VAC	**Voltage Alternating Current**
VRV	**Variable Refrigerant Volume**
XMPP	**Extensible Messaging and Presence Protocol**
WiFi	**Wireless Fidelity**
WLAN	**Wireless Local Area Network**
WMP	**Wireless Message Protocol**
WP	**Work Package**
ZIGBEE	**Zonal Intercommunication Global-standard**

1 INTRODUCTION

1.1 Scope and objectives of the deliverable

The goal of the D3.1 deliverable is to present the activities conducted within T3.1 as of M10 of the project. Furthermore, this deliverable provides a thorough description of the components that constitute the D^2EPC IoT Framework responsible for the real-life information collection and cleansing from the project's demonstration buildings. The D^2EPC IoT framework steps on the solution provided by Hypertech, customize it, and further enhance it to fulfill the project requirements. More specifically, it aims at providing seamless and secure communication with smart metering and sensing equipment deployed on-site at the pilot installations of the project, or interfacing with existing infrastructure, when possible, to ensure the availability of adequate real-life information necessary for the definition of dynamic and context-aware EPCs.

The dynamic metric (T2.2/3/4) and system operation (T2.5) requirements, along with the analysis of both the existing infrastructure (defined under T5.2) and the areas of focus of the pilot tests, will shed light on specific installation needs for each of the demonstration buildings. As the works of the respective tasks have not yet concluded, the deliverable includes a high-level introduction of the pilot sights. It further provides available information prior to the submission of the current document concerning the existing metering and sensing equipment as well as the respective data retrieval methods that will be utilized to ensure constant and secure communication with the components of the IoT Framework. The second version of the deliverable will include the finalized information regarding the pilot audits as well as a dedicated chapter to the data requirements defined within the tasks of WP2.

D3.1 further provides insights on the IoT Equipment Framework. It consists of the Wireless Sensor Network, which comprises the sensing/metering devices installed in the building, and the IoT Gateway, a smart bridge that gathers the information from the installed devices. The selection of the sensors, meters, and gateways depends on a set of evaluation criteria that ensure the smooth integration of the IoT equipment. The IoT Equipment Framework components and evaluation criteria along with the IoT implementation process are described within the first version of the deliverable. The second and final version of the deliverable will include the works conducted to define an indicative list of off-the-shelf devices to be deployed in the pilots.

From the software point of view, D3.1 presents the components that constitute the Information Management Layer. The IML is responsible for collecting, processing, and transmitting the data streams from the installed IoT devices to the project's common repository. Moreover, it is responsible for interfacing with building management systems in order to utilize existing metering/sensing infrastructure in the building. The deliverable provides a description of the IML components and examines the interactions of the IML within the D^2EPC system architecture. The next version will include the standard information interfaces between the IML and the interacting D^2EPC components as well as the interfaces with the BMS infrastructures available in the pilot sights.

Lastly, two Annexes (A and B) are included in the current document that provide insights on Hypertech's solution and one Annex (C) that provides detailed specifications of the existing pilot equipment. Annex A focuses on the IoT Gateway. It contains the hardware specifications of the IoT Gateway along with the software definition of the internal subcomponents responsible for several applications, such as IoT device integration, interfacing with the IML, commissioning etc. Annex B provides a detailed description of the IML components responsible for processing and handling the data streams and the components responsible for interfacing with the IoT Gateway or external software components.

1.2 Structure of the deliverable

The deliverable content is separated into five (5) chapters analyzing the topics mentioned in the introduction. A brief description of the sections and their respective content is presented below:

- **Chapter 1 – Introduction:** This chapter provides an introduction to D3.1, its objectives and scope of work, as well as its relation to other deliverables.
- **Chapter 2 - Building Case Studies & IoT Infrastructure Requirements:** This chapter aims at offering an overview of the pilot site hardware and software infrastructure availability and reveal their remaining metering and sensorial equipment requirements.
- **Chapter 3 – IoT Equipment Framework:** This chapter aims at describing the hardware aspects of the D^2EPC IoT Framework. The current version of the deliverable introduces the IoT Gateway, which is the central unit of the D^2EPC IoT Framework. Within Chapter 3, the Gateway role in the D^2EPC information flow and its respective capabilities and the conceptual representations of its software subcomponents are presented.
- **Chapter 4 – IoT/IML CLOUD SPECIFICATIONS:** This chapter describes the software aspects of the D^2EPC IoT Framework, introducing the IML Cloud component responsible for processing the data collected and streaming them to the project repository. This chapter presents not only the IML Cloud technological capabilities but also its respective subcomponents and describes its functionality within the D^2EPC framework based on the D^2EPC Architecture.
- **Chapter 5 – Conclusions:** This chapter concludes the overall work conducted in T3.1 and provides an overview of next steps and the content of the next version of the deliverable.

1.3 Relation to Other Tasks and Deliverables

Considering the fact that T3.1 aims to deliver the entire IoT Framework, a number of interdependencies arise with multiple other tasks of the D^2EPC project. More specifically:

- **WP1:** The backbone of the D^2EPC project is the system architecture which will be the outcome of **T1.4**. The architecture defines the interactions between the individual components of the project to enable their exclusive development. Hence, the building blocks of the IoT framework and its interactions with other D^2EPC components were determined under **T1.4**.
- **WP2:** WP2 will deliver the necessary data requirements for the evaluation and operation of the system within D^2EPC. More specifically, the works conducted in **T2.1**, **T2.2**, **T2.3,** and **T2.4** will elicit a set of key performance indicators to cover the novel aspects of the dynamic certificate. The respective metrics required for the calculation of the KPIs will provide feedback to T3.1 in terms of mandatory IoT equipment. Moreover, **T2.5** is responsible for the definition of the typical information model, which will further deliver data requirements concerning the operation of the system's tools.
- **WP4:** In order to enhance the information credibility of the collected data streams, **T4.3.2** will deliver the Energy Performance Verification & Credibility tool, which is part of the Extended D^2EPC application toolkit. The Information Management Layer (IML) component will stream data and network flows to the tool; thus, the necessary interfaces need to be determined.
- **WP5:** WP5 is responsible for the preparation of the pilot sites and the deployment of the D^2EPC solutions. Therefore, the interaction of the T3.1 with the WP5 activities is bidirectional. Initially, the audits performed under **T5.2** will provide extensive information concerning the existing infrastructure and characteristics of the demonstration buildings highlighting potential connectivity issues or feasibility constraints. T3.1 will contribute with a tailored list with proposed off-the-shelf IoT devices, which will set the grounds for the definition of the IoT equipment to be installed at the pilots (bill of materials, **T5.2**), and its deployment plan (**T5.3**).

2 Building Case Studies & IoT Infrastructure Requirements

2.1 Introduction

D^2EPC aspires to demonstrate the developed technologies and solutions in six (6) diverse demonstration case studies in Germany, Cyprus, and Greece. These are briefly presented in the following paragraphs to introduce the project's pilot sites. A more thorough analysis of these case studies will be covered in D5.3 "Pilot Planning and Setup v1", which is expected on M18. The summary of the existing infrastructure at the presented sites in section 2.1.2 includes all information available when submitting the current document (M10). The finalized version summarising the complete information regarding the IoT infrastructure and interfaces available at the pilot sites will be provided in the second version of the deliverable (M16).

2.1.1 Summary of pilot infrastructure

2.1.1.1 Case Study 1 – nZEB Smart House DIH, Thessaloniki, Greece

This case study consists of the CERTH/ITI nZEB Smart House at CERTH premises in Greece. It is a 316 m^2 rapid prototyping demonstration infrastructure shaped like a real residential household. The main section of the building represents a single-family, detached residential building and is already equipped with many IoT smart solutions that provide much information about its operational characteristics. The building is divided into two main sections, the household section and the control/demo rooms. The former is mainly equipped with an extensive IoT infrastructure, whereas the latter has fewer sensors/systems deployed. It is envisioned that the entire building will be used for demonstrating the D^2EPC solutions; however, the final IoT topology will be determined in the second version of the deliverable (M16).

It is worth mentioning that the building is also equipped with a 9.57kWp PV system and a 5kWh Li-on battery, which makes it an ideal case study for exploring prosumer scenarios.

Figure 1. Case Study 1 - nZEB Smart House DIH, Thessaloniki, Greece

2.1.1.2 Case Study 2 - Residential/ Multi-family building, Velten, Germany

The building, constructed in 1907, consists of six apartments (three floors) and a basement where all the mechanical equipment is located. Currently, the building is occupied by ten (10) tenants, and the main assets that are of interest are the heating and domestic hot water (DHW) systems. These assets were installed in 2009 and use natural gas as the primary source of fuel. The coverage of the building for the D^2EPC project is two apartments, which are located on the second and third floors. The building is equipped with wireless sensors that enable the close monitoring of air quality and environmental conditions. Currently, a smart meter has been installed and is under testing. This meter will deliver energy consumption in kWh in near real-time, among other energy-related parameters. A weather station, which is planned to be installed in the vicinity of the building, will collect a wide range of data from barometric pressure to wind speed, among others.

Figure 2. Case Study 2 - Residential/ Multi-family building, Velten, Germany.

2.1.1.3 Case Study 3 – Industrial/Tertiary building, Berlin, Germany

Case study 3 concerns a 2235 m^2 building, constructed in 1925, with four different heating installations, two gas-powered HVAC systems, one gas-powered base building heating system, and one gas-powered "black heater" in the production area. The building is used by a metal working company and can be divided into the following areas: the Stainless-steel production hall 2 also hosting a plasma cutting machine, the production hall 1, the staff room, the work equipment warehouse, the lathe and milling shop, the polish & paint shop and a warehouse. On the first floor, above the work equipment warehouse and turning and milling shops, are the company's offices. The other areas are designed as industrial halls. The different areas of the building also have different insulation and heating and will be addressed accordingly through the D^2EPC framework.

Figure 3. Case Study 3 – Industrial / Tertiary building, Berlin, Germany

2.1.1.4 Case Study 4 – Mixed-use University building, Nicosia, Cyprus

The building in Cyprus is located in plots 512 and 513, Part 01, Sheet/Plan 21/470201 (Longitude and Latitude 33°22'46.70 "E, 35°10'46.20"N, Link), in the area of Palouriotissa, Nicosia (Y. Frederickou Str.). Frederick University's new wing building is a two-story 2100m^2 building, its volume is approximately 7,100m^3 (included the basement floor/parking area), and it was built in 2007. The understudy building does not border with any other building. The building consists of a basement (area of 450m^2), ground floor (area of 545m^2), and two floors (area of 545m^2 each floor). University's cafeteria is on the ground floor; on the first floor, there are three seminar halls of 220 students capacity, and offices are found on the second floor. The building can host up to 390 people. The total height of the building is 15.60m from the basement floor to the terrace. The individual heights of the floors are 4.10m for the typical floors and the ground floor. The services that are provided within the building include heating, cooling, ventilation, lighting, and electrical appliances. A 3kW Solar Water Heating System is currently installed on the building.

Figure 4. Case Study 4 – Mixed-use University building, Nicosia, Cyprus

2.1.1.5 Case Study 5 – Multi-family Apartment Building, Berlin, Germany

The original plan for this case study includes a classical multi-family building located in the center of Berlin, initially constructed in 1900. This building is already equipped with smart meters and sensors, and its energy performance and consumption are currently monitored in order to be optimized. Therefore, historical energy data are available. The overall living surface of the building is approximately 1685 m^2. As a historical building constructed in the early 1900s, several restrictions should also be taken into account when considering energy performance improvements measures and renovation scenarios.

The building has four floors and consists of 16 apartments hosting 41 occupants in total. It is also equipped with a solar thermal system and a PV system.

Figure 5. Initial Case Study 5 –Multi-family Households Building, Berlin, Germany

The demonstration cases 5 & 6 remain to be finalized and might be updated in the future, given ongoing discussions regarding the feasibility and optimal deployment of the D^2EPC framework. Based on the data requirements to be defined in WP2, different buildings may be selected instead.

A good alternative proposed to replace or complement the initial pilot site suggestion is an apartment building erected in 1996. This building consists of 18 flats. The tenants are already participating in an energy-monitoring scheme. The building is heated via methane, has a green roof, and a solar thermal installation on the roof at the southwest oriented yard side.

Figure 6. Alternative Case Study 5 – Multi-family Households Building, Berlin, Germany

2.1.1.6 Case Study 6 – Multi-family Apartment Building, Berlin, Germany

The last pilot building, Case study 6, is also located in the center of Berlin, and it is a classical multi-family building. The building was initially constructed in 1911 and constituted a historical building as well. It consists of four floors, and overall, there are 12 apartments, with 36 occupants of mixed age. The envelope of the building consists of exterior walls made from brickwork and wooden windows, providing an ideal case study for renovation. The building is already equipped with smart devices, and its performance is being monitored, allowing for tackling integration aspects.

Figure 7. Case Study 6 – Multi-family Households Building, Berlin, Germany

2.1.2 Pilot information flow

The pilot information collection described below corresponds to the ongoing auditing process conducted at the D^2EPC demonstration buildings until M10. The respective technical specifications of the equipment can be found in Annex C. In the next version of the deliverable to be released on M16, the complete information concerning the existing equipment and acquisition methods will be enriched and presented in detail.

2.1.2.1 Case Study 1 – nZEB Smart House DIH, Thessaloniki, Greece

As a demonstration site for R&D purposes, this building is equipped with a variety of IoT sensors and systems. Divided into two main sections, that due to the different usage will be considered as two separate thermal zones, the infrastructure deployed covers indicatively the following aspects:

- Indoor Temperature, Humidity, and Luminance per space (for the house section)
- Occupancy (presence/absence, number of people, exact location)
- HVAC system operational status
- Lighting system operational status
- Outdoor temperature, wind speed, and direction, rain concentration, solar irradiation (through a roof weather station)
- Water Meters (central supply, grey water supply, domestic hot water through a solar water heater)
- Inverter Status and Measurements (for the PVs and ESSs)
- Electricity Meters (central and five internal ones, as well as dedicated to the HVAC external units)

The infrastructure mentioned above communicates with different protocols and gateways to an IoT Platform available for real-time monitoring and control. The covered protocols are wired (BACnet, Modbus, Canbus, etc.) and wireless (ZigBee, EnOcean, Z-wave, BLE, LoRa, etc.), offering a highly diverse ecosystem for testing purposes. To support all these protocols, the infrastructure is well equipped with the necessary infrastructure (in the form of gateways), also allowing for easy extensions and scalability scenarios. In some cases, there are two or more alternatives for measuring similar aspects, but from different positions. An indicative list of available devices is presented in the list below.

The IoT Platform that collects and handles this data is also equipped with an API that offers either RESTfull services or event-based communication through MQTT or XMPP protocols. For the D^2EPC purposes, this case study can offer direct communication to the IoT infrastructure where possible or through the API offered by the existing IoT Platform.

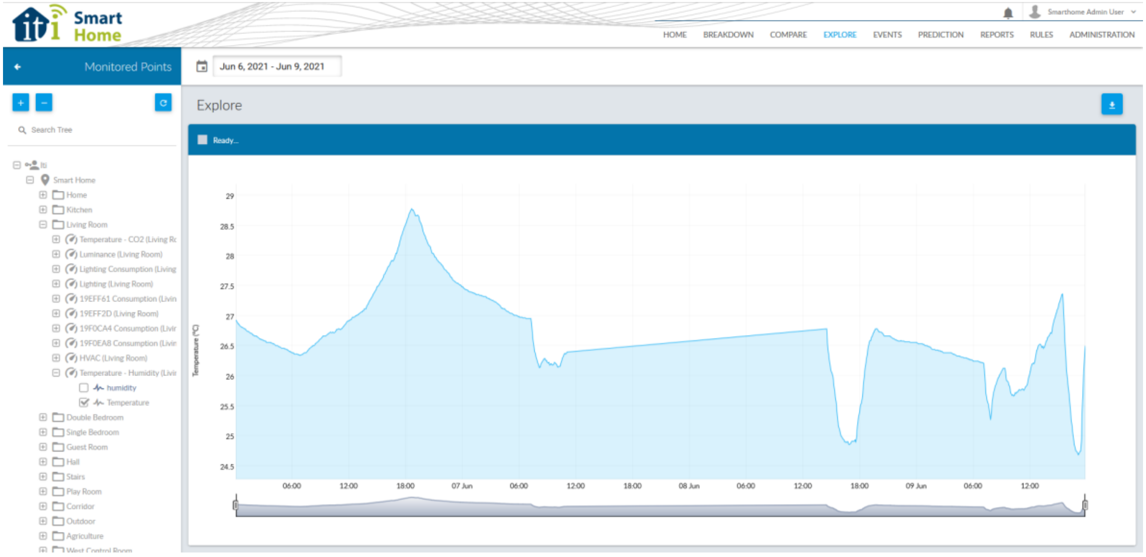


Figure 8. nZEB Smart House IoT Platform

Currently, the communication protocol used by the IoT Platform does not follow a specific standardized approach (i.e., specific data model, sequence, etc.), but rather a custom JSON-oriented data format. However, this is something that can be adjusted if needed to facilitate integration activities.

An example of the simplified JSON payload for a temperature/humidity sensor and the "last value" measurement is provided below:

Last Value Payload

```
{
  "temperature": 24.7546,
  "humidity": 44,
  "timestamp": "2021-06-09T13:52:43.796Z"
}
```

The frequency of the data originating from this highly diverse IoT ecosystem varies significantly. Some indicative examples are the electricity meters, which are from Carlo Gavazzi, and allow for a resolution of one second through Modbus RTU. In contrast, for indoor conditions, one example would be the Plugwise sense that sends a measurement over ZigBee every 15' (adjustable).

Table 1. Devices installed in the nZEB Smart House

Type of device	ID number	Image	Location	Connectivity	Data Access	Measuring interval
TEMPERATURE & HUMIDITY						
Plugwise Sense	-		One per each space in the main House	ZigBee HA 2.0	SmartHome IoT Platform API	15' -adjustable
Gavazzi EM340	-		One per indoor electric panel (3 ground floor)	Modbus RTU	SmartHome IoT Platform API	1 sec from device 1 min from API
Gavazzi EM270	-		Building Point of Common Coupling	Modbus RTU	SmartHome IoT Platform API	1 sec from device 1 min from API
Thermokon SR04 CO2	-		1 living room ground floor and 1 playroom first floor	EnOcean		**Measuring interval** WakeUp time = 100 sec. (default) **Transmission interval** every 100 sec. at change >0,4 K, >2,5% rH or 50 ppm, otherwise every 1000 sec.

2.1.2.2 Case Study 2 - Residential/ Multi-family building, Velten, Germany

This demonstration site aims to present the integration of two apartments. Currently, the building is equipped with temperature, humidity, and CO_2 sensors deployed inside the apartments and the staircase of the building. Table 2 provides detailed information on the modules installed in this building.

Additional equipment is expected to be installed through the D^2EPC project covering energy monitoring meters (currently, one smart meter has already been installed for testing purposes, and a second one is planned to be installed after successful testing and having the specifications from WP2) and a weather station (the station has been purchased and is also under testing in the vicinity of the building).

The existing sensors communicate wirelessly through Sigfox, whereas the smart meter uses NBIoT technology, and wireless communication is also being considered for future installations. The devices installed in the building do not require WLAN. The information collected through the sensors is managed by an online web IoT platform that enables device management, data collection, processing, and visualization. Also, it supports communication with third-party tools through a REST API call node – see Figure 9.

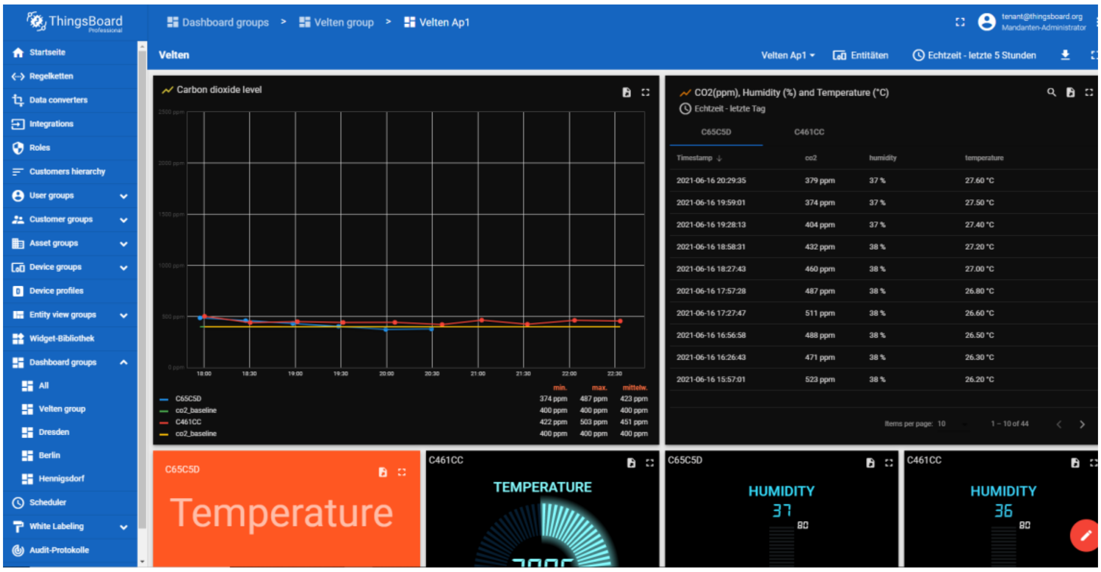


Figure 9. CLEOPA's IoT Platform

If needed, for the integration with the D^2EPC Information Management Layer, a dedicated API is envisioned to allow for robust data extraction from the building. Figure 10 illustrates an example of payload parsing for modules measuring CO_2, temperature, and humidity.

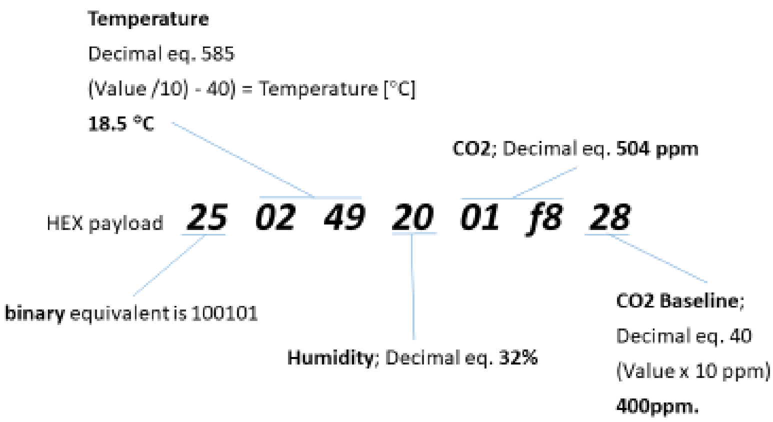


Figure 10. Byte analysis for sensor payload from module measuring CO_2, temperature, and humidity

Example of Sigfox callback - JSON Body:

{
"data":"{data}",
"time":"{time}",
"device":"{device}",
"TEMP":"{customData#Temperature}",
"CO2":"{customData#CO2}",
"HUM":"{customData#Humidity}"
}

Table 2. Devices installed in the Velten building

Type of device	ID number	Image	Location	Connectivity	Data visualization/ download	Measuring interval
TEMPERATURE AND HUMIDITY						
Airwits R4.1	3357B2		Velten, hall	Sigfox	ThingsBoard	30 min measuring interval –adjustable
Airwits R4.1	335340		Velten, hall	Sigfox	ThingsBoard	30 min measuring interval –adjustable
CO2, TEMPERATURE AND HUMIDITY						
Airwits CO2 R5.2	C461CC		Velten, Apt. 1 2nd Floor	Sigfox	ThingsBoard	30 mins measuring interval –adjustable
Airwits CO2 R5.2	C65C5D		Velten, Apt. 1 3rd Floor	Sigfox	ThingsBoard	30 mins measuring interval –adjustable
Airwits CO2 R4	2F117B		Velten, Apt. 2 2nd Floor	Sigfox	ThingsBoard	30 mins measuring interval –adjustable
Airwits CO2 R4	3A6F24		Velten, Apt. 2 3rd Floor	Sigfox	ThingsBoard	30 mins measuring interval –adjustable
TOTAL NUMBER OF DEVICES		6				

Regarding the measurement and transmission of data, the default setting is 30 minutes, but it can be adjusted to 10 minutes, 20 minutes, one hour, or two hours.

The smart meter installed is a DINRail 3-phase meter and measures the following parameters:

Electrical parameters measured*	Irms, Vrms, Active Power Reactive Power and Energy Consumption in kWh
Ranges of measured parameters	Voltage: 0 to 285 Vac between phase and Neutral Current: up to 600Amps
Tolerance	+10% of the nominal load (Iov)

Accuracy of measurements	1% of reading measurement error (**)

** The meter can measure and transmit data every one minute (this is the minimum report interval). It measures power factor (PF) and also provides information on harmonics.*

*** Accuracy refers to electric power measurements*

The electricity meter enables remote monitoring and controlling a household's energy consumption and/or an industrial building. It is a wireless metering system that uses NB-IoT technology for communication, which is the most widely applied technology for smart metering. One of the significant advantages of NB-IoT is its deep penetration compared to other technologies. This makes it suitable for the case of Germany, considering that several buildings have thick walls that are difficult to penetrate (particularly to the basement). The device has internal data storage, which is suitable for data collection also when unexpected network connectivity errors from the provider side may occur. Therefore, the collected data is not lost.

2.1.2.3 Case Study 3 – Industrial / Tertiary building, Berlin, Germany

Although the conduction of the demo activities of D^2EPC at the production halls is primarily explored, other industrial areas will also be examined for inclusion extending the envisioned IoT infrastructure. Currently, the building is equipped with temperature, humidity, and CO_2 sensors deployed on the production and paint halls. Additional equipment is expected to be installed through the D^2EPC project. Table 3 provides detailed information on the modules installed.

Existing sensors communicate wirelessly through Sigfox. The information collected through the sensors is handled by an online web platform, supporting communication with third-party tools through a REST API.

If needed, for the integration with the D^2EPC Information Management Layer, a dedicated API is envisioned to allow for robust data extraction from the building. Figure 11 illustrates an example of payload parsing for modules measuring CO_2, temperature, and humidity.

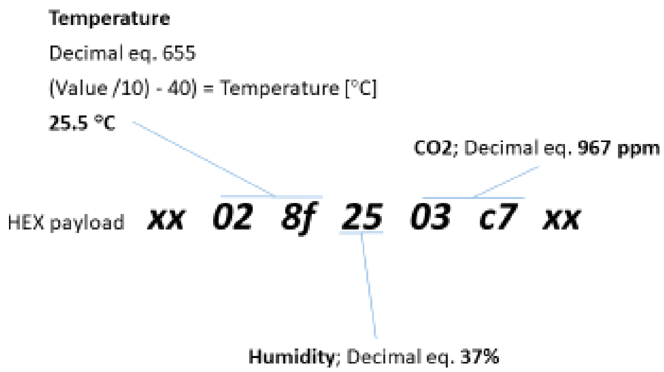


Figure 11. Byte analysis for sensor payload from module measuring CO_2, temperature, and humidity, the second example

The sensors currently available in this building are listed in Table 3.

Table 3. Devices installed in the Berlin building

Type of device	ID number	Image	Location	Connectivity	Data visualization/ download	Measuring interval
TEMPERATURE AND HUMIDITY						
Airwits R4.1	335894		Berlin Ernst & Freyer - thermal cutting hall.	Sigfox	ThingsBoard	30 min measuring interval –adjustable
Airwits R4.1	34AD98		Berlin Ernst & Freyer - working hall.	Sigfox	ThingsBoard	30 min measuring interval –adjustable
CO2, TEMPERATURE AND HUMIDITY						
Airwits CO2 R5.2	C660BF		Berlin Ernst & Freyer - production area.	Sigfox	ThingsBoard	30 mins measuring interval –adjustable
Airwits CO2 R5.2	C65C54		Berlin Ernst & Freyer - thermal cutting hall.	Sigfox	ThingsBoard	30 mins measuring interval –adjustable
TOTAL NUMBER OF DEVICES		**4**				

Concerning the information sharing method, it is the same as in the Velten case.

2.1.2.4 Case Study 4 – Mixed-use University building, Nicosia, Cyprus

The building introduced in this case study is also a multi-use building with quite a diverse set of spaces, systems, and assets. The entire new wing building covered is divided into three separate zones, which will be monitored in detail. The entire building will also be covered in terms of energy monitoring, providing a complete data flow that fully depicts the building's status. The installed equipment is connected to a system allowing monitoring, control, and remote sensing of the actual energy performance of the building, as well as enabling the realization of the dynamic EPC scheme.

In more detail, the infrastructure deployed covers indicatively the following aspects:

- Temperature
- Humidity
- CO_2
- Outdoor temperature, wind speed, and direction, rain concentration, solar irradiation (through a roof weather station)
- Energy Meters for electricity consumption

The above equipment communicates through different protocols, such as BLE and WiFi, with all the information being collected into two central data logging devices. These devices are equipped with integrated web servers, allowing for remote communication, monitoring and control through web interfaces. These interfaces will be used for integrating to the D^2EPC information management layer, collecting all necessary information for performing the assessment foreseen.

The installation of equipment follows different characteristics per floor, as depicted below.

- **Ground floor**

The canteen area is an open space. Therefore, all the illumination is measured in one circuit. A second circuit exists for the kitchen appliances (refrigerators, ovens, etc.), a third circuit for the entrance area, and a fourth circuit for the toilets. The elevator is measured from the basement (or from the roof).

- **First floor**

There are three seminar rooms, a corridor and two toilets. The measurements involve the lights and the sockets in each room, the hallway lights, and the toilet lights on this floor.

- **Second floor**

The offices are on this floor, divided into five zones; four with cabinets (lighting, sockets), and one for the conference rooms. Apart from that, there is also a zone for the corridors and a zone for the toilets.

- **Roof**

On this floor, there are five measurements in five VRVs, as well as circumferential lighting (one circuit).

The abovementioned information is summarized in the following table:

Table 4 Equipment installations per floor in the mixed-use University building

Zone	Lighting	Sockets	A/C	Total
Ground floor	x 4 (canteen area, kitchen area, entrance, toilets)	x 3 (canteen area, kitchen area, entrance)	N/A	7 or 8
First floor	x 5 (3 seminar rooms, entrance, toilets)	x 4 (3 seminar rooms, entrance)	N/A	9
Second floor	x 7 (3 office wings, seminar rooms, entrance, toilets)	x 6 (3 office wings, seminar rooms, entrance)	N/A	13
Roof		N/A	VRV x 5	6

The 3-zone monitoring and remote sensing, as well as the power consumption of the building, are implemented by using the equipment shown in the following table:

Table 5. 3-Zone monitoring and remote sensing equipment in the mixed-use University building

Model	3-Zone environmental monitoring and remote sensing description
MX1102A	HOBO Carbon Dioxide - Temp - RH Data Logger Bluetooth Low Energy (BLE)
MXGTW1	MX Gateway (Included Service Plan: MX Gateway HOBOlink Data Plan for the first year)
SP-620	MX Gateway HOBOlink Data Plan (3 x annual data plan for each Gateway)
Dataloggers	**Energy performance of the building description**

Ground floor:	
EG4115 Pro (Onset 10744)	30 Input Meter Data Logger
First floor:	
EG4115 (Onset 10740)	15 Input Meter Data Logger (EG4115 Core Data Logger)
Second floor:	
EG4130 (Onset 10712)	15 Input Meter Data Logger (EG4130 Core Data Logger)
Roof:	
EG4115 (Onset 10739)	15 Input Meter Data Logger (EG4115 Core Data Logger)

2.1.2.5 Case Studies 5&6 – Multi-family Apartment Building, Berlin, Germany

These case studies target to demonstrate the rollout for dynamic energy performance certificates based on BIM and near-real-time measurements data from sensors and smart meter readings for energy, district heating, and gas. Focusing on the building's measurements , an initial proposal covers the following infrastructure/interfaces:

- Meters for energy consumption (electricity, fuel, solar, etc.)
- Average room temperature for a significant room to see whether there is a bias in room temperature
- Average CO_2 level in the rooms, to have an indicator for the air infiltration rate

The building's owner has offered to allow access to data collected by a subcontractor, modulating set temperature to reduce energy demand. The data stream of the meters of the buildings is to be connected to HTTP://www.energiesparkonto.de.

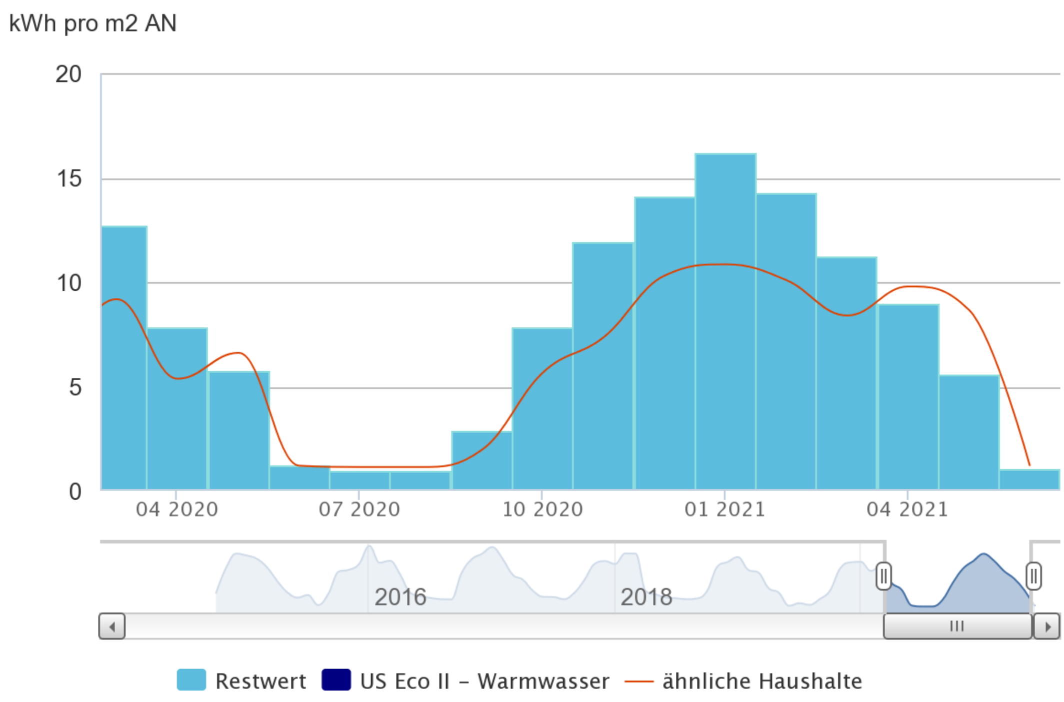


Figure 12. Meter data storage portal.

The information for consumption-based information via an API will be derived as shown in Figure 13.

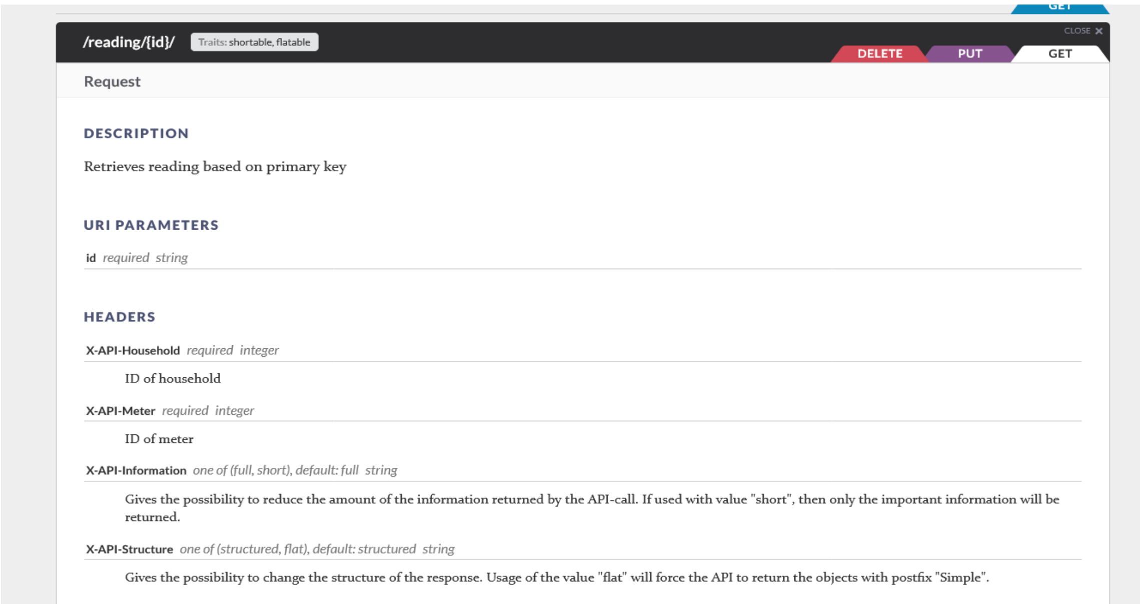


Figure 13. API for case studies 5/6

In Berlin, the district heating operator also offers smart meters as shown in Figure 14, but the access to the collected information is not confirmed at the moment.

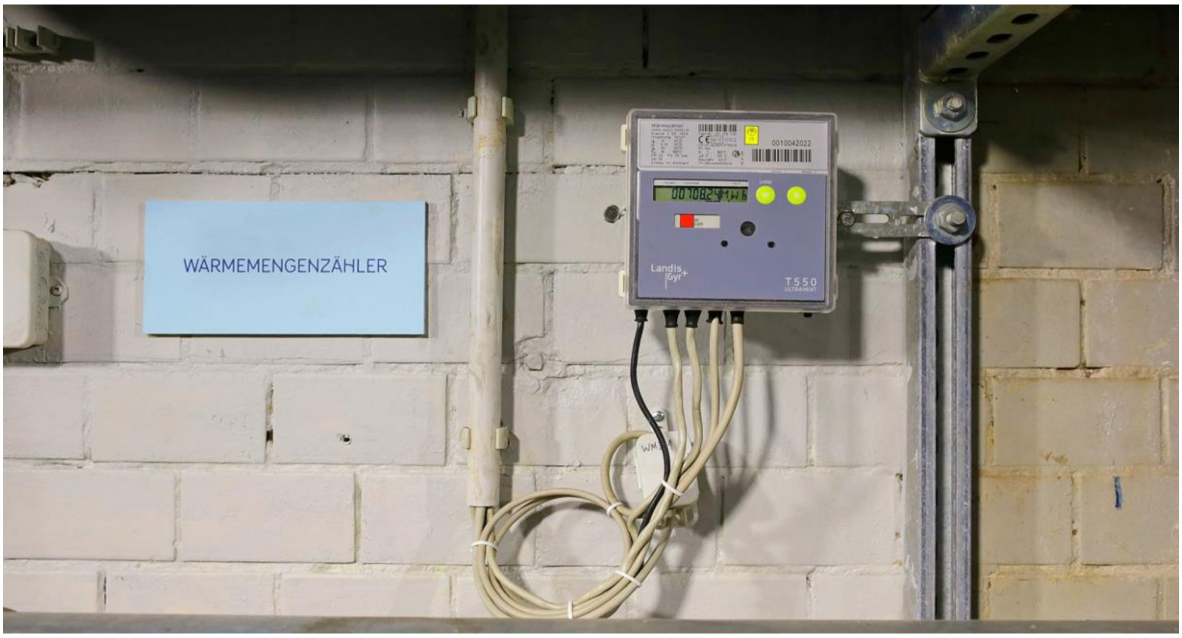


Figure 14. Smart meter district heating

Additionally to the building data, an interface to the weather service is employed using CSV download for the degree day correction. Retrieving those data from central D^2EPC components will also be possible via an API. Occupation and used floor area necessary to calculate operational consumption bound EPC will be entered via web form.

3 IoT EQUIPMENT FRAMEWORK

D^2EPC aims at launching a new foundation of energy performance certificates that utilize state-of-the-art information technologies and expand their field of interest to other concepts beyond energy consumption. The next-generation digital and dynamic EPCs will be based on a set of novel key performance indicators to examine the building in terms of smart readiness, human comfort/well-being, environmental impact, and operational cost. The certificate will further introduce advanced building information modelling by adopting the "digital twin" concept, a simulated real-life representation of the building. The simulation will be designed to display asset consumption and indoor conditions at the lowest possible spatial granularity taking as input the constant flow of data from metering equipment inside.

D^2EPC tools and services will be extensively tested in realistic terms in the pilot sights that will participate in the trials. Towards achieving this, a smart device network will be deployed at the pilot sites to collect real-life information, including but not limited to: energy consumption of specific loads, areas of the building or total consumption, indoor ambient conditions, weather data, operational statuses etc. Infrastructure already available on-site, including installed sensors, meters and/or SCADA systems, will be utilized provided that interfacing with the D^2EPC framework is possible. This re-utilization of already available infrastructure is desirable not only on the grounds of cost and time efficiency considerations but also to reduce nuisance to the end-users and avoid interacting with the space aesthetics through the excess device deployment.

In order to retrieve the (near) real-time information seamlessly and securely, D^2EPC will utilize Hypertech's IoT solution and evolve it to update and enhance its functionalities to fulfill the project requirements. This solution will be tailored to address the project's pilots' characteristics, existing infrastructure, and specific features. The IoT Framework will operate as a smart building monitoring system that collects the information from the sites, cleans and verifies it, and proceeds with forwarding the data streams to the D^2EPC repository to be available to other project components.

The IoT Equipment Framework can be conceptually considered as two separate components. The **Wireless Sensor Network (WSN)** comprises multiple hardware devices (smart meters and sensors) installed in the pilots and the **IoT Gateway,** which ensures the interoperability of the WSN in a protocol-agnostic manner and facilitates continuous communication with the **IML Cloud.** The latter is responsible for processing and streaming information to the project's common repository (as described in the project's system architecture [1]).

The WSN enables the inclusion of a wide range of IoT devices & commercial communication protocols in order to establish a system capable of:

i. capturing the indoor ambient conditions from the pilot site corresponding (but not limited) to temperature, occupancy, illuminance, humidity, air quality, or the outdoor conditions (such as weather data)
ii. recording the total electricity consumption of the demonstration buildings as well as the consumption of specific loads by implementing an energy metering setup (meters/sub-meters)
iii. including smart devices already available on-site through a protocol-agnostic manner

The IoT Gateway is the system component equipped with the necessary functionalities that enable the end-to-end interoperable communication between the installed IoT devices and other components of the D^2EPC. More specifically:

- The **bridge functionality** allows the integration of multiprotocol IoT devices delivering a common interface to the information management layer.

- The **gateway functionality** enables the constant and uninterrupted data to flow from the wireless sensor network to the IoT cloud.

The IoT Gateway developed by Hypertech is a cost-efficient innovative bridge -with extended autonomous sensor/actuator capabilities- that delivers bidirectional communication between the WSN of the buildings and the IML cloud. It is designed to integrate a multitude of off-the-shelf IoT devices communicating through well-known and established communication protocols. In the physical world, the Gateway corresponds to a compact and discreet device hosted inside the pilot premises. It runs on a Raspbian operating system which incorporates the software applications that materialize the functionalities mentioned above. It is already applied, tested, and described in several EU projects; therefore, the complete description of its software applications, conceptual representations, and hardware specifications are included in Annex A.

3.1 Evaluation of the IoT Equipment Framework

3.1.1 Challenges in implementation and evaluation criteria of the Wireless Sensor Network

Previous experience has shown that a plethora of obstacles occur during the definition and deployment of the Wireless Sensor Network in pilot premises. The list of challenges presented below has been identified during both the system development and testing activities of past projects and are associated with several factors [2]. More specifically:

- **Compatibility & Interoperability:** With an increasing number of vendors, OEMs, and service providers, maintaining interoperability between different IoT systems can be a challenging process. The abundance of novel features and implementation strategies of communication standards increases the complexity of device integration exponentially.
- **Privacy & Security:** Beyond the complexity considerations, integrating multiple IoT devices on an IoT platform might result in several types of security risks. A compromised system may pave the way for jeopardizing the network's health and safety, leading to potential data theft, loss of privacy, and eventually non-compliance with laws and regulations governing the data collection, storage, and distribution.
- **Connectivity:** Achieving seamless communication among the sensor network and other components of the IoT Framework, a stable wireless internet connection is considered a prerequisite to eliminate the inconsistencies in data capturing and enable quick and quality communication.
- **Scalability:** Adding a large number of IoT devices (i.e. nodes) to an IoT framework to address project needs might lead to the system’s rough operation, slowing its performance.
- **Intrusiveness:** The most suitable IoT solution for metering or sensing does not always comply with the users’ needs. Many users raise objections to certain devices due to practical or aesthetic reasons.
- **High Power Consumption:** Constant data exchange at a high frequency is a power-demanding procedure. If the device is not adequately optimized (both hardware and software-wise), overheating issues might occur. Such issues lower the device’s durability, which ultimately leads to malfunctions in operation.
- **High Cost:** The finalized equipment deployed in a demonstration building might comprise a multitude of IoT devices. Due to the limited resources available per project, high-cost devices - even with desirable characteristics - might be excluded from the selection.

- **Availability:** A research for the optimal device solution should always consider the availability in terms of quantity or time of delivery.

The abovementioned challenges have been heavily considered in order to facilitate the system implementation, speed up the device installation and integration process and satisfy the occupants' needs with the least possible intrusiveness. A specialized set of criteria has been adopted to determine and optimize the selection of IoT equipment. The finalized list of devices to be installed in the D^2EPC demonstration buildings are expected to fulfill the criteria presented below:

- **Reliable Solution:** A reliable device functions appropriately for a more extended period of time. In order to ensure uninterrupted data collection at the pilot sites, the equipment to be selected will be widely tested, certified and highly rated.
- **Secure Solution:** State-of-the-art and certified security technologies and mechanisms employed to eliminate the likelihood of data theft or loss of privacy, in line with the EU regulations.
- **Interoperable Solution:** Every solution integrated into the D^2EPC IoT Framework is extensively pre-tested in terms of compatibility with the existing infrastructure prior to deployment.
- **Plug and Play Device:** The involvement of non-expert stakeholders and commissioners makes the ease of installation and configuration of the IoT equipment highly desirable features. Plug and play devices are designed based on user-friendliness, hence are generally promoted.
- **Support of widespread communication protocols:** In order to satisfy reliability, security, and interoperability requirements, all devices communicate with well-known and established communication protocols.
- **Mature Solution:** A previously adopted solution by a large number of users or stakeholders provides blueprints for future installations. Feedback by an active community - concerning a specific device - may significantly shorten the debugging process.
- **Off-the-shelf Device:** The term “off-the-shelf” implies that the device is a well-known solution readily available in the market. Such devices are commonly designed to serve the needs of a wide range of users, maintaining a reasonable cost while being readily available, reducing delays in the procurement process.

3.1.2 Evaluation of the IoT Gateway

Due to the diverse characteristics of the demonstration buildings, the involvement of a diverse group of users and use cases, and the need for reduced nuisance and aesthetics disruptions, the selection of IoT Gateway is determined following four principal pillars:

- **Adaptability**: The solution needs to fulfill the universal applicability requirement in order to satisfy the end-users needs in every pilot premise.
- **Acceptance:** The solution is expected to reach a certain level of acceptance by the users in terms of ease of use and installation as well as aesthetics.
- **Modularity:** A modular solution offers maximum device manageability and minimum complexity in the context of implementation and maintenance.
- **Reasonable Cost:** The final installation topology might impose the installation of multiple gateway devices. As a result, the device cost must be maintained at a reasonable level.

3.2 Implementation Process of the IoT Equipment Framework

The specifications of the finalized IoT Equipment Framework to be implemented in the demonstration buildings will be based on the activities conducted within WP2 and WP5. The ongoing auditing process (WP5) at each pilot site will determine the building's structure and characteristics, technical building systems, the loads and assets to be measured, and the pre-existing metering and sensing devices. In parallel with the works of WP5, the set of key performance indicators and the common information model will be defined by the works carried out by WP2. The KPIs' calculation methodologies will provide feedback on the metrics that need to be measured in the pilots along with the respective data frequency and the spaces of interest. Further data requirements related to the operation of the system tools will result from the definition of the D^2EPC data model.

The WP2 and WP5 activities outcome will shed light on what needs to be measured, what is already measured, and how these data streams will be acquired from the pilot sights. This information will provide the feedback necessary for the final definition of the requirements in terms of hardware and interfaces to be established to extract data from the systems available on site. Therefore, the second version of the current deliverable will conclusively define an indicative list of IoT devices in line with the project needs and specifications to be used as an input for the WP5 pilot site deployment activities.

Towards eliminating potential hardware and software compatibility issues, the IoT equipment will be assessed in terms of interoperability, functionality, and operation for the smooth-running establishment of the IoT network. Moreover, the considered hardware will be assessed in terms of cost, intrusiveness, and user acceptance.

Each recommended metering or sensing device will be extensively pre-tested under lab or real-life conditions. The lab trials will be performed locally at Hypertech's premises which are appropriately configured to act as an IoT testing facility. In cases when a corresponding pilot infrastructure is not available in Hypertech's test bed and acquiring it is not an option due to techno-economic considerations, the tests will be performed remotely in close collaboration with the pilot partners responsible. The lab or remote trials will set the grounds for defining the finalized Bill of Materials for each pilot site in T5.2. Within the second version of the *IoT Platform & Interfaces* deliverable (D3.4), all the works conducted during the testing procedure in order to determine a proposed list of IoT devices for the formation of the BoM will be documented in detail.

4 IoT/IML CLOUD SPECIFICATIONS

4.1 Introduction

The vast amount of information coming from the wireless sensor network leads to cumulated volumes of data that raise new challenges concerning their manageability, quality, and secure transmission. To address these issues, Hypertech provides specially designed components to gather, process, and distribute data between the building blocks of D^2EPC in a timely and cost-effective manner.
The Information Management Layer is Hypertech's cloud-based component responsible for administrating the sensing and metering datasets extracted from the Wireless Sensor Network deployed at the pilot premises. To accomplish this, the IML comprises the different software elements necessary for handling the collected data streams and interfacing between the modules of the project. The IML component is therefore designed and developed in such a manner as to address various challenges related to big data directly linked to the IoT domain [3]. More specifically:

- **Volumes of Data:** IML sets its grounds on cloud computing, enabling on-demand availability of computer system resources in terms of data storage.

- **Security & Privacy:** Communication between the IML and other components is realized through the HTTPS protocol maximizing the security of data exchange. All data collected is anonymized following the EU GDPR, and to ensure maximum privacy, the access to data is strictly limited.

- **Transmission Speed:** The solution is designed on the basis of modern, optimized, and open-source software components that minimize latency.

- **Selective Data Acquisition:** The recording strategy of data is event-based; hence, a new value is only registered if it fluctuates compared to the latest update. This type of filtering minimizes the overall storage load without significant loss of information. Furthermore, it reduces the number of messages sent, hence the number of operations which ultimately leads to higher speed and lower power consumption.

- **Scalability:** The IML permits the registration of multiple pilot premises and prosumers to maintain high standards in terms of effective operation.

- **Data Extraction:** All datasets within the IML are transformed into a common structured format (JSON), facilitating the data extraction and analysis.

The role of the D^2EPC IML is not limited to simply collecting information coming from the Wireless Sensor Network and forwarding it to the D^2EPC repository. The component implements well-known and established algorithms responsible for cleansing and normalizing the various datasets related to energy consumption and ambient conditions. Furthermore, it is equipped with specialized tools capable of displaying and monitoring the respective datasets. The overall solution provides a high level of:

- **Data Quality:** The processing algorithms eliminate the inconsistencies and discontinuities in the vast sensorial and metering datasets and ensure that high-quality information is forwarded to the data-driven components.

- **Data Visualization & Monitoring:** The respective tools are optimized to deliver quality visualizations in a timely manner facilitating the user to identify patterns and monitor events within the collected datasets. In the context of D^2EPC, the visualization tool is not foreseen to be utilized. However, it will be offered to the pilot partners to facilitate the maintenance of the pilot sight equipment.

Hypertech's IML component is a mature solution already applied and tested in various EU projects. Its functionalities concerning the data processing and the interfacing between the IoT Gateway and other external components have been previously documented and found in Annex B of the deliverable. D3.1 also covers the role of the IoT Framework - and more specifically, the IML - within the D^2EPC system architecture. The second version of the deliverable will describe integrating the Information Management Layer component to the overall D^2EPC platform. More specifically, it will provide detailed documentation of all the necessary interfaces between the IML and the shared repository of the project that will be defined in collaboration with the works of T2.5. Furthermore, it will include the work conducted (in collaboration with T4.3) to define the IML interfaces with the second interacting component, the Energy Performance & Credibility tool.

4.2 D^2EPC IML Cloud and Information Management

4.2.1 D^2EPC IML Cloud

The software building blocks and interactions of the IoT Framework with other components of the D^2EPC are described within the System Architecture of the project. Figure 15 presents a UML diagram with the D^2EPC Conceptual Architecture segmented in layers. More specifically:

Physical or Infrastructure Layer: In D^2EPC, the Wireless Sensor Network resides in the physical layer of the system architecture. The layer contains the physical world devices deployed locally at the pilot sites, including smart sensors and meters or other infrastructure available on site, such as weather stations. Several demonstration buildings are equipped with building management systems (BMS) or SCADA systems to gather information from existent metering/sensing devices. In order to obtain these measurements, the existing infrastructure is either commissioned to the IoT Gateway or integrated with the IML through specially defined interfaces. Lastly, external sources of information, such as weather APIs can be integrated into the IML to collect necessary data and forward it to the D^2EPC repository.

Interoperability Layer: The Information Management Layer (IML) resides in the interoperability layer and consists of the subcomponents designed to conduct critical tasks related to data extraction/collection and data processing. Namely, the three components comprising the IML are:

- The **Application Layer** incorporates all the algorithms responsible for cleansing, normalizing, and transforming the datasets in order to provide high-quality information in the appropriate format to other D^2EPC components. Additional processing necessary for the D^2EPC component operations such as aggregation is also conducted within the Application layer.
- The **IoT Interfaces** functions as an intermediary between the IoT Gateway and the IML, providing the necessary interfaces for continuous and secure communication between the two.
- The **Common Information Interfaces** act as a mediator facilitating the information exchange between the IML and other D^2EPC components or services.

At this point, it is worth noting that the **IoT Gateway** lies beneath the physical and interoperability layer, assuming a twofold role.

- Administration of the wireless sensor network which corresponds to the integration and data collection of the IoT devices (physical layer)
- Establishment of the communication interfaces with the interoperability layer. This corresponds to a common interface with the IML through the protocol translation between the IoT Gateway and the multiprotocol IoT devices.

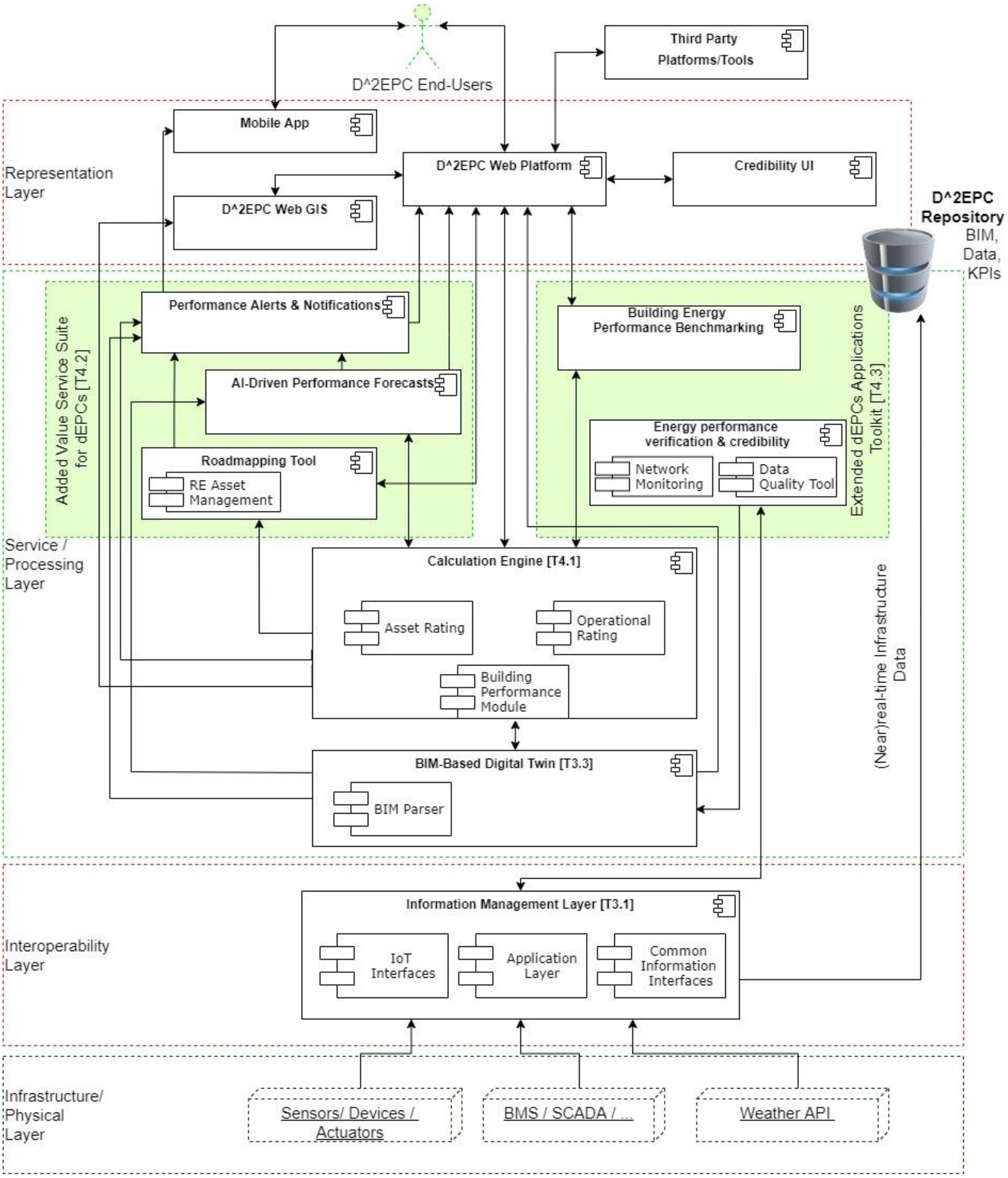


Figure 15. D^2EPC Layered Conceptual Architecture [1]

As presented in Figure 15 15, the IML Cloud component interacts only with two modules of the D^2EPC architecture; the D^2EPC repository and the Energy Performance Credibility & Verification module. No other interfaces with the rest of the D^2EPC modules have been foreseen. This happens because the data extracted from the locally deployed wireless sensor network is transmitted from the IoT Gateway to the IML. After being cleaned and processed, the data is streamed to the **D^2EPC common repository.** It remains available to all the tools and services of the project that require real-life data during their operation.

The second component interacting with the IML is the **Energy Performance & Verification**. Its purpose is to ensure the reliability and credibility of the acquired data from the Wireless Sensor Network. This happens by applying an automated procedure that evaluates the collected information and identifies irregularities related to the data patterns that IML's cleansing mechanism could not have detected. Given that the objective of the specific module is relevant to the acquisition of high-quality data, the specific module interfaces directly with the IML to ensure the timely detection of problematic data sets and flag them before transmitting them to the D^2EPC repository. In Figure 16, a functional UML diagram highlights the information flow between the IML Cloud and the respective interacting components. [1]

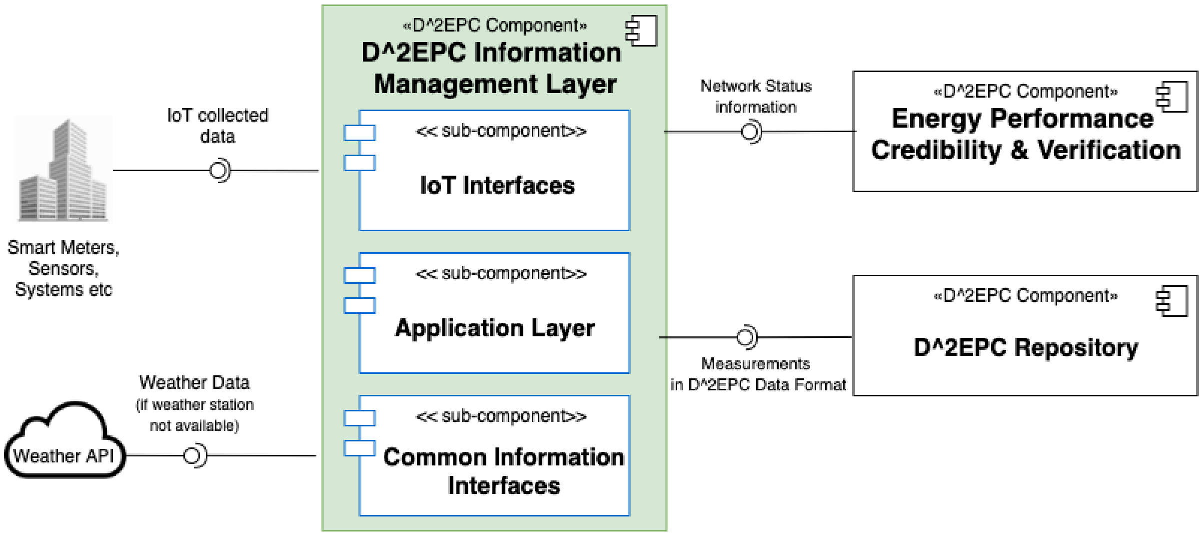


Figure 16. D^2EPC Information Management Layer UML functional diagram

5 CONCLUSIONS

The first version of D3.1, ”IoT *platform & Interfaces*” describes the followed methodology that will deliver the complete D^2EPC IoT Framework upon the completion of T3.1. Within D3.1, the already available information from the findings of the audits as of M10 is included. The description of the pilot sights allows an overview of their topology and limitations. At the same time, the high-level summary of the available equipment and interfaces provides invaluable information on the existing infrastructure. It sheds light on the additional IoT equipment required to be deployed per pilot.

The deliverable also examines the challenges that arise from an IoT Framework implementation and provides a holistic overview of the framework itself utilized to overcome these challenges. The IoT Framework components correspond to the Wireless Sensor Network, the IoT Gateway, and the Information Management Layer. The former two components represent the physical world sensing/metering devices and the bridge device deployed locally at the pilot sites. The latter (IML) is the software component responsible for the data collection and processing. D3.1 further describes the role of the IoT Framework within the D^2EPC system architecture, summarising work from D1.4 and offering further details related to the component itself, such as its subcomponents and functionalities.

The final version of the deliverable will conclude the works of T3.1. The final dynamic metrics requirements and the summary of the complete outcomes of the pilot audits will be documented. In addition, the lab and remote tests (if necessary) results will also be discussed towards the definition of an exhaustive equipment list to be used for the definition of the pilot sites’ BoMs. The tests will be conducted to identify the applicability of the IoT equipment considered for the project demo activities. Lastly, T3.1 will describe in detail the interfaces established between the IML and the infrastructure available at the D^2EPC pilot sites.

References

[1] CERTH, GSH, DMO, HYP, AEA, SEC, FRC, "D^2EPC Framework Architecture and specifications v1," 2021.

[2] A. &. K. S. Khanna, "Applications and Challenges: A Comprehensive Review," Wireless Personal Communications, pp. 1687-1762, 2020.

[3] J. K. V. P. a. A. D. P. Wongthongtham, Big Data Challenges for the Internet of Things (IoT) Paradigm, 2017.

ANNEX A: IoT Gateway Specification

The IoT gateway is designed to act as a bridge between the physical environment and the computer-based systems. In principle, the developed solution is hosted by a Raspberry-pi device. It incorporates several software components developed to establish uninterrupted and secure communication between the WSN installed locally and the cloud-based IML. The overall software stack consists of two layers and several applications responsible for network configuration, security, and updating, presented in Figure 17. Each component and its functionalities are described in detail in the following sections.

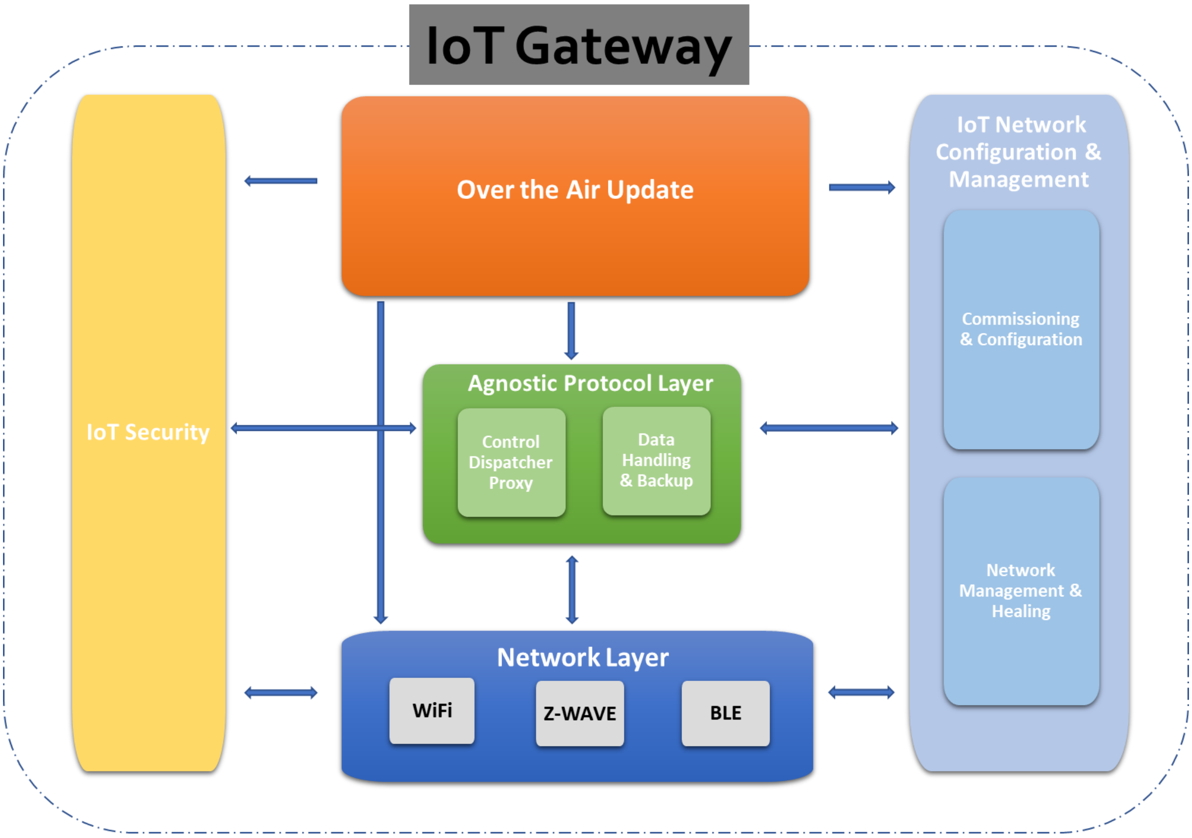


Figure 17. Conceptual representation of software components hosted on the IoT Gateway.

A.1 IoT Gateway Software Definition

A.1.1 The Network Layer

The Network layer is responsible for the network connectivity of the Gateway and the data exchange with the sensor network on site. The Network layer handles the streaming of operational status information and sensing data from the corresponding IoT devices. The principles and functional characteristics upon which the Network layer was designed and developed are summarised in Table 6.

Table 6. Network Layer Specifications.

Network Layer Specifications	
System robustness/Error reporting	To minimize the error probability of the installed IoT devices (such as device malfunction, power breakage, connection loss etc.) the network layer requires a management system that identifies and responds to network status deviations. It is also acting proactively by regularly executing diagnostic tests.
Low energy consumption	To ensure device durability, the network needs to maintain a low energy consumption profile. To achieve this, the battery-powered IoT devices considered for deployment are communicating through protocols that allow secure and low power communication
Scalability	The scalability of the Network layer is of great importance to enable the integration of a large number of installed devices while ensuring their uninterrupted and error-free operation. System scalability is subject to the network limitations related to the utilized protocols within the network layer. Such limitations can be overcome by installing extra gateways in the areas/spaces of focus inside the building

The network layer further includes subcomponents that facilitate communication with the external IoT devices. Each subcomponent corresponds to a communication protocol commonly used by off-the-shelf smart devices, including the WiFi manager, the Z-Wave manager, and the BLE manager. These components evolve and extend through the system testing and as the need to include additional IoT devices increase.

A.1.2 The Agnostic Protocol Layer

The Agnostic Protocol Layer is a protocol-agnostic building automation platform that runs as the center of the smart building solution. Its objective is to enable interoperability by integrating a multitude of smart devices and systems, such as other well-known home/building automation solutions, (smart) devices and sensors, and other technologies (ModBus TCP etc.), into a single solution. Regardless of their communication protocol in a uniform and protocol-agnostic manner. In addition to commercial protocols, the agnostic protocol layer is designed to integrate tailored controllers to communicate with legacy building assets, not supporting network interfacing; for example, legacy AC units allowing IR-based remote control.

The software solution of the agnostic protocol layer is based on openHAB 2.4[1] , an open-source application. openHAB was selected based on its significant benefits, most important of which being:

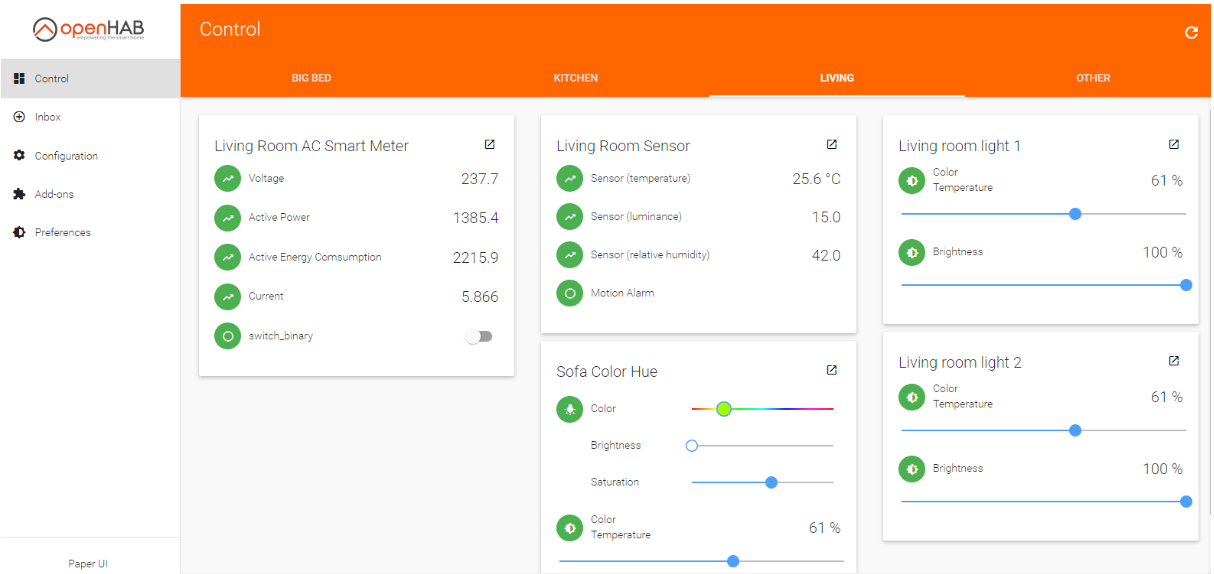
- Wide support of different technologies (and consequently IoT devices) and potential for further expansion to additional devices
- Vendor and platform neutrality
- Runs on any device compatible with Java Virtual Machine

[1] https://www.openhab.org/

- Availability of a rule engine that provides triggering upon events
- Provision of a user-friendly interface (PC, mobile) for managing dashboards, reports, configurations, and benchmarks
- Well documented and an active technology community providing support

The basic components of the openHAB's data model[2] that represent the functional view of the physical devices are:

- **Things**: As described by their name, Things represent home automation entities (devices, web services, or other sources of information and functionality) that can be added to a system. openHAB can interact or extract data from a Thing (e.g. temperature sensor, smart light bulbs etc.) through **Channels**. Every Thing in openHAB is linked to one or more Channels to enable the interaction between them (e.g. for an energy meter, the available channels could be power, voltage, current etc).
- **Bindings**: A Binding is an add-on software component that enables the communication between openHAB and a Thing by omitting any communication-specific requirements of the Thing to handle it in a protocol-agnostic manner. Through Bindings, openHAB can integrate physical hardware, external systems, and web services (e.g. Philips Hue binding, z-wave binding etc).
- **Items**: Items represent all capabilities of a Thing that can be either used by a UI or in Automation logic (e.g. device ON/OFF status, sensor temperature etc). **Links** act as the connector between one Channel and one Item



. openHab control screen.

openHAB offers two different internal communication channels, an **asynchronous event bus,** and a **REST API**.

The **event bus** is responsible for the asynchronous communication between two components upon an event. More specifically, the supported communication events are sent and received through the event bus-only upon their occurrence. Some examples of these events are provided below:

[2] "Concepts." https://www.openhab.org/docs/concepts.

- **Item Commands Events** which trigger actions or state change of an item (e.g. light is switched on, setpoint temperature is increased),
- **ItemStateChangedEvents** indicating that the state of an Item has changed (e.g. temperature increases more than 0.01ºC), and
- **ThingStatusChangedEvents** indicating status changes of the connected Things (e.g. a sensor going offline)

The openHAB **REST API** is an HTTP interface that grants access to all data related to Items, Things, and Bindings, enabling their configuration or the influence of other openHAB elements.

As far as the bindings responsible for the communication between the openHAB-based agnostic protocol layer and the IoT devices considered are concerned, both existing and internally developed bindings have been used. An exhaustive list is presented below:

- **Z-wave binding**[3]: As several IoT devices selected communicate through a z-wave protocol, this binding covers the majority of z-wave supported devices. For z-wave devices that were not recognized by the z-wave binding during the lab testing and could not be automatically traced by the agnostic protocol layer, their configuration was performed manually. The existing z-wave binding was updated to recognize them in the future.
- **MQTT Binding**[4]**:** This binding establishes a link between the available openHAB items and the MQTT message topics defined under the project requirements.
- **Intesis Binding:** This custom-made binding enables different actions related to the openHAB communication with the HVAC gateway, including automatically discovering such devices available in the local Wi-FI network, establishing a TCP connection to the detected devices, exchanging TCP telegrams under the WMP communication protocol[5].
- **Modbus Binding**[6]**:** This binding enables access to Modbus TCP and serial slaves, supporting communication with different Modbus variants such as RTU, ASCII, and BIN.
- **Phillips Hue & Ikea Tradfri Bindings**[7]**:** Two bindings relative to the control of widely commercial lighting fixtures have been included. The first one is associated with the Philips HUE light fixtures and the (IKEA) TRÅDFRI binding. Both bindings enable communication to the commercial gateways related to each type of light fixture.

Complementary to the above list, another openHAB component responsible for data storage in openHAB can be retrieved later on for different purposes (e.g. restore your system after startup) corresponds to the **Persistence services.** The respective services utilized within Hypertech's solution are listed below:

- **Real-time monitoring and reporting on the device status**: Suitable for fault reporting and network mapping

[3] "ZWave Binding." https://www.openhab.org/addons/bindings/zwave/.

[4] "MQTT Binding" https://www.openhab.org/addons/bindings/mqtt/

[5] "WMP Protocol Specifications." https://cdn.hms-networks.com/docs/librariesprovider11/manuals-design-guides/wmp-protocol-specifications.pdf?sfvrsn=339b5cd7_6

[6] "Modbus Binding. " https://www.openhab.org/addons/bindings/modbus

[7] "Philips Hue Binding." https://www.openhab.org/addons/bindings/hue/

"TRÅDFRI Binding." https://www.openhab.org/addons/bindings/tradfri/

- **External Service binding:** Extra bindings can be added that enable communication with different devices and services (such as Nibe Cloud, Fronius Inverters, and others).
- **MQTT Persistence Service:** This service allows to feed item states to an MQTT broker using the openHAB persistence strategies

The agnostic protocol layer further comprises two subcomponents providing extra functionalities regarding control dispatching and data backup. More specifically:

- **Control Dispatcher Proxy**: This module enables the dispatching of manual or automated control commands to the concerned actuators.
- **Data Handling and Back-up**: This subcomponent gathers sensing/metering information from the sensor network and forwards it to the IML cloud. Due to a number of reasons, including security issues, device health, and memory limitations, no raw data is stored locally in the Gateway apart from the core information required for specific functionalities. As an exception to that, a backup mechanism has been established, which temporarily retains the collected data in cases of network disruptions until the gateway communication is restored and the information is successfully uploaded to the cloud server. In this way, information loss that can arise from communication disruptions is prevented.

A.1.3 Over the Air Update

The *Over the Air Update* application is responsible for updating the firmware in an over-the-air manner of all other components and subcomponents which reside within the IoT Gateway

A.1.4 IoT Security

This component is responsible for protecting private and sensitive data exchange and storage through authentication and authorization techniques. Relevant privacy and security protocols and standards are considered and established within the security module, which interacts with every other component and subcomponent of the IoT Gateway.

A.1.5 IoT Network Configuration and Management

The IoT Network Configuration and Management includes two subcomponents; the Commissioning & Configuration and the Network Monitoring & Healing subcomponent. Below, a brief description of each subcomponent is provided:

- **Commissioning & Configuration:** This application is responsible for the installed devices' discovery and commissioning. It provides the commissioner with an interface enabling a step-by-step procedure to register the IoT devices to the building's automation system. This subcomponent acts as a support tool facilitating quick and easy IoT equipment deployment.
- **Network Monitoring & Healing:** This application enables network status monitoring to detect and solve emerging errors and timely communication disruptions. This subcomponent aims at facilitating the maintenance of the locally installed IoT equipment.

A.2 IoT Gateway Hardware Set-up

The IoT Gateway is based on Raspberry Pi 4 Model B (Figure 18), a low-cost and small-sized solution powerful enough to host the software components of the bridge. Raspberry-Pi 4 offers a plethora of functionalities providing a complete operating system and additional development environment capabilities. The device is fully capable of simulating a conventional desktop PC, using a standard keyboard, mouse, and plugging into a computer monitor or TV. The Raspberry Pi 4 Model B has a built-in 1.5GHz, 64-bit quad-core processor, 2GB SDRAM memory, multiple port support (USB 2.0/3.0, micro-HDMI, microSD), and broad connectivity (wireless LAN 2.4/5 GHz, Bluetooth, BLE). A USB-C power supply powers the raspberry board. Detailed specifications are presented in Table 7. Due to the sensitivity of its hardware parts, the Raspberry PI 4 Model B is enclosed in a suitable case to both protect the board and offer added value by aesthetically enhancing the Gateway (Figure 18).

Figure 18. (left) Raspberry Pi 4, (right) enclosure case example.

Table 7. Technical specifications of Raspberry PI 4 Model B.

Processor	Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
RAM Memory	2GB LPDDR4-3200 SDRAM
Wireless Connectivity	2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
Input/Output	Raspberry Pi standard 40 pin GPIO header
Ports	2 USB 3.0 ports; 2 USB 2.0 ports.
	2 × micro-HDMI port

	2-lane MIPI DSI display
	2-lane MIPI CSI camera
	4-pole stereo audio and composite video
	Micro-SD card slot for loading operating system and data storage
Power	5V DC via USB-C connector (minimum 3A*)
	5V DC via GPIO header (minimum 3A*)

The Raspberry Pi is further enhanced with a 32GB SD card to increase its storage capacity as well as a z-wave card to facilitate communication with z-wave devices. The Z-Wave Plus Aeotec Z-Pi 7 daughter card sits on top of the Raspberry PI GPIO connector and turns every Raspberry PI into a Z-Wave Home Automation Gateway. The respective Z-Wave software provides a user-friendly interface utilizing the JavaScript object notation (JSON) format on a built-in web server. The integrated Z-wave antenna allows the Gateway to broadcast and receive signals at a range of up to 200m. Detailed specifications are presented in Table 8.

Figure 19. (left)32 GB Micro SD, (right) Z-Wave Plus Aeotec Z-Pi 7 card.

Table 8. Technical specifications of Z-Wave Plus Aeotec Z-Pi 7 card.

Protocol	Z-Wave Plus
Frequency	868.4 MHz

Range	up to 100m indoors / up to 200m outdoors
Dimensions	42 x 35 x 12 mm
Power Supply	GPIO DC 3,3V

ANNEX B: IML Cloud Components

B.1 Introduction to the IML Cloud

The Information Management Layer Cloud component is responsible for collecting real-time information related to the pilot buildings. The component guarantees a secure environment for processing the vast amount of data streamed through the IoT gateway and collected from the installed IoT device network. Furthermore, it enables communication with external cloud components for data and information exchange. The IML Cloud comprises four subcomponents described briefly below:

- **Data Processing** is the subcomponent where the collected data streams are processed through the implemented cleansing and transformation techniques.
- **Data Handling** is the subcomponent where information is retrieved or transmitted to the Gateway through the IoT Interface component.
- **Common Information Interfaces** refers to the subcomponent where data exchange with other cloud or software components is realized.
- **IoT Interface** is the subcomponent where messages are transmitted to the respective application component within the IoT Gateway, providing a reliable and secure communication framework.

In Figure 20, the UML component diagram illustrates the subcomponents - and their interactions - included in the IML cloud. Each subcomponent presented in this diagram is described in detail in the following sections.

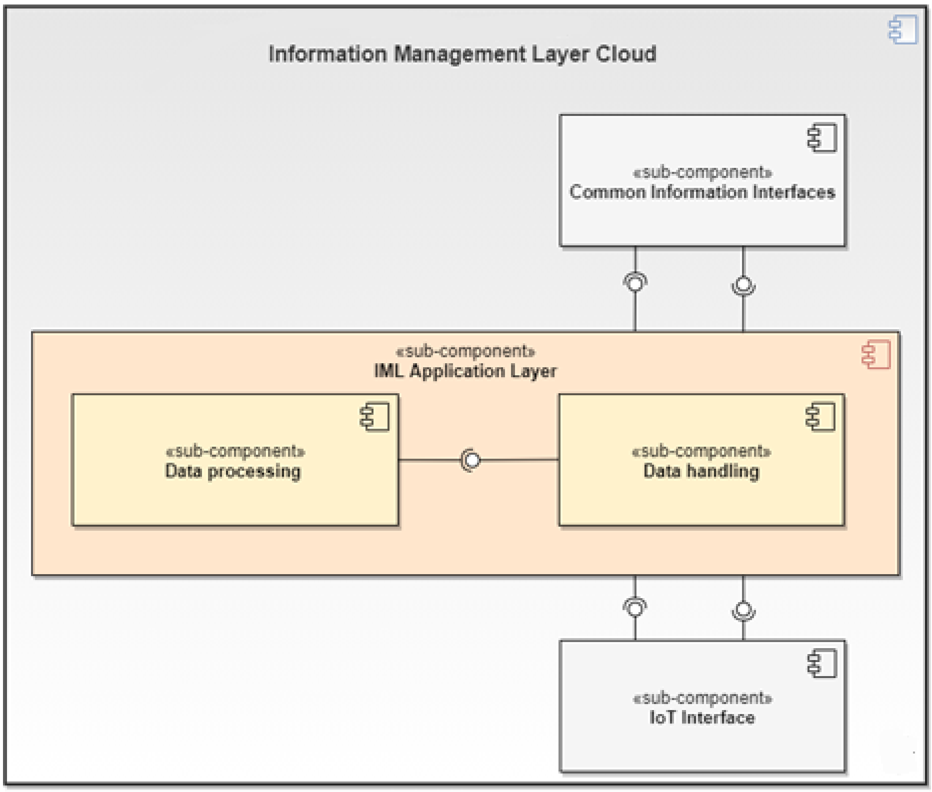


Figure 20. IML Cloud UML component diagram.

B.2 Data processing (Cleansing/Visualization)

The Data Processing Tool is the IML Cloud subcomponent responsible for cleansing and transforming the vast amount of data acquired from sensors (ambient conditions) and meters (power/energy consumption, operational status etc.). The unified algorithmic framework includes two methods:

- Data Cleansing
- Data Transformation

B.2.1 Data Cleansing

The modern data-driven modelling techniques are highly dependent on the quality level of the time series data collected by the IoT devices and systems. Values that significantly differ from the values generally assumed by the same metric are considered anomalies that may lead data-driven models to draw erroneous conclusions if retained in the data set. Thus, it is mandatory to apply specialized cleansing techniques in order to eliminate such anomalies and enable other processing tools to extract the correct information from the measurements of the sensor network.

The data cleansing tool developed and implemented within the data processing component of the IML cloud is responsible for:

- **Outlier Detection:** Appliance of methods that detect values that lie at a significant distance from other values in a random metric sample (outliers)
- **Outlier Treatment:** Appliance of methods that treat outliers by removing them and imputing relevant data values

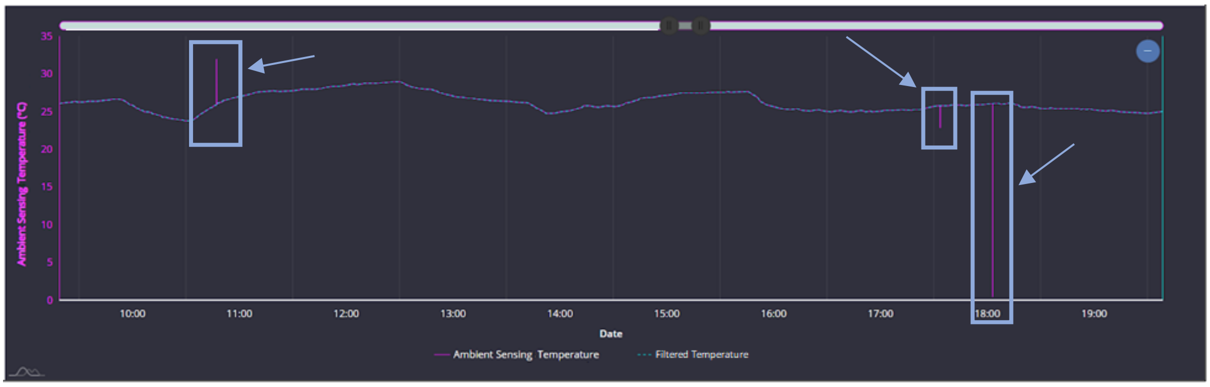


Figure 21. Outlier values on an ambient sensing temperature profile.

B.2.1.1 Outlier detection methods

In Hypertech's cleansing approach, the outliers are determined without prior knowledge of the data other than the type of measurement. The values within the dataset are not pre-characterized as outliers or normal (unlabelled data). Thus, the detection procedure is based on an unsupervised learning approach. The statistical and cluster-based methodology employed in the cleansing tool depends on the various profiles of the metrics measured by the IoT devices. A different algorithmic approach is adopted per metric type. After extensive testing and assessment of several applicable algorithms' performance, it has been selected based on their success rate (% of outliers detected) and their computational time (elapsed runtime). The algorithms finally chosen to be included in the cleansing tool are the non-parametric Moving Window Filtering, Hampel Filter and DBSCAN, and the parametric Grubb's test. In the following section, a brief description of each algorithm is provided and the results from the lab trials on various sensing/metering metrics.

Moving Window Filtering Algorithm (MWFA)[8]:

The MWFA is a non-parametric statistical method that utilizes the expected value (mean) and standard deviation of a moving window in order to filter out undesired values in a dataset. More specifically, the algorithm scans the dataset in rolling intervals, including k number of values (window length), and for each interval, the mean and standard deviation is calculated. If the absolute difference between a value within a window and the window's mean exceeds a certain threshold (determined by the standard deviation of the window), then the corresponding value is considered to be an outlier. This can be mathematically expressed as follows:

$$|p_i - \bar{p}_i(k)| < as_i(k) + \gamma = \begin{cases} True, & observation\ i\ is\ kept \\ False, & observation\ i\ is\ substituted \end{cases}$$

Where:

$\boldsymbol{p_i}$ is the value under examination

$\boldsymbol{k}$ is the heuristically predefined length of the window (window size)

$\boldsymbol{\bar{p}_i}$ is the mean value of the moving window

$\boldsymbol{s_i}$ is the standard deviation of the moving window

$\boldsymbol{a}$ is the heuristically predefined number of standard deviations to act as a threshold

$\boldsymbol{\gamma}$ is a heuristically predefined parameter that helps to avoid zero variances produced by sequences of k equal values

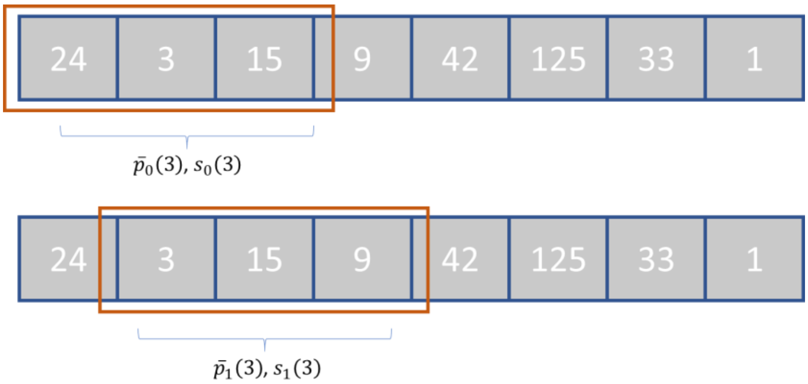


Figure 22. Moving window representation with window size equal to 3.

The MWFA performed more satisfactorily in terms of both outlier detection effectiveness and computational time on metrics following fairly regular patterns where consequent measurements tend to present small percentage variations in temperature and humidity. In Figure 23, a raw temperature data set is compared with the resulting cleansed data set after applying the MWFA. The graph inside

[8] Chawalit Jeenanunta, K. Darshana Abeyrathna, M. H. M. R. Shyamali Dilhani, Su Wutyi Hnin1 and Pyae Pyae Phyo (2018) Time Series Outlier Detection for Short-Term Electricity Load Demand Forecasting. International Scientific Journal of Engineering and Technology (ISJET), Vol. 2 No. 1

the red frame represents the rescaled version of the raw values graph after the exclusion of the outliers.

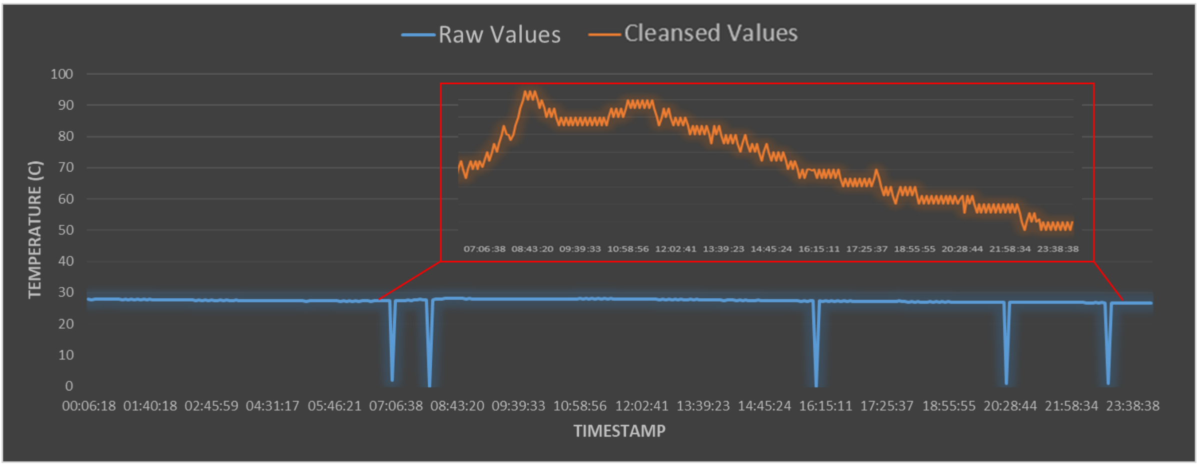


Figure 23. Moving window representation with window size equal to 3.

Hampel Filtering[9]:

The Hampel filtering is a non-parametric statistical method analogous to the MWFA, based on the median and Median Absolute Deviation (MAD). For each instance of the input signal, MAD is calculated for a window composed of the current data value and (k-1)/2 adjacent values (where k is the window length). If the value differs from the window median by more than the predefined threshold, the filter replaces the sample with the median. This can be mathematically expressed as follows:

$$p_n = \begin{cases} p_n, |p_n - \widetilde{p_n}(k)| \leq tMAD_n(k), observation\ n\ is\ kept \\ \widetilde{p_n}, |p_n - \widetilde{p_n}(k)| \geq tMAD_n(k), observation\ n\ becomes\ the\ median \end{cases}$$

Where:

$\boldsymbol{p_n}$ is the value under examination

$\boldsymbol{k}$ is the heuristically predefined length of the window (window size)

$\widetilde{\boldsymbol{p_n}}$ is the mean value of the window

$\boldsymbol{MAD}$ is the median absolute deviation of the window

$\boldsymbol{t}$ is the heuristically predefined number of median absolute deviations to act as a threshold

[9] Ronald K. Pearson* , Yrjö Neuvo, Jaakko Astola, Moncef Gabbouj (2016) Generalized Hampel Filters. Pearson et al. EURASIP Journal on Advances in Signal Processing, 2016:87 DOI 10.1186/s13634-016-0383-6

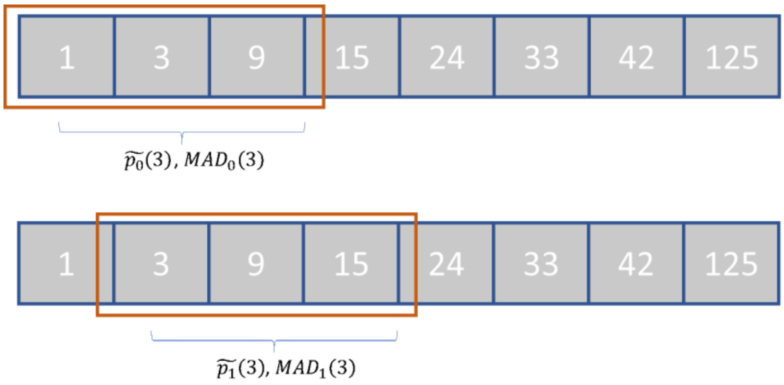


Figure 24. Hampel filter representation with window size equal to 3.

The Hampel filtering achieved a higher score than MWFA on metrics containing value ramps (like the cumulative energy consumption). In such cases, the utilization of the median, used in Hampel filtering instead of the mean used in the MWFA, facilitated the detection process since the former is insensitive to the outliers. However, in large samples, a significant increase in the Hampel filtering algorithm runtime was observed. In the majority of cases, the MWFA performed equally well in terms of accuracy at a lower computational cost.

Figure 25 presents the results of the Hampel filter cleansing method applied to energy consumption data. The cleansed data visualization reveals the cumulative nature of the energy consumption.

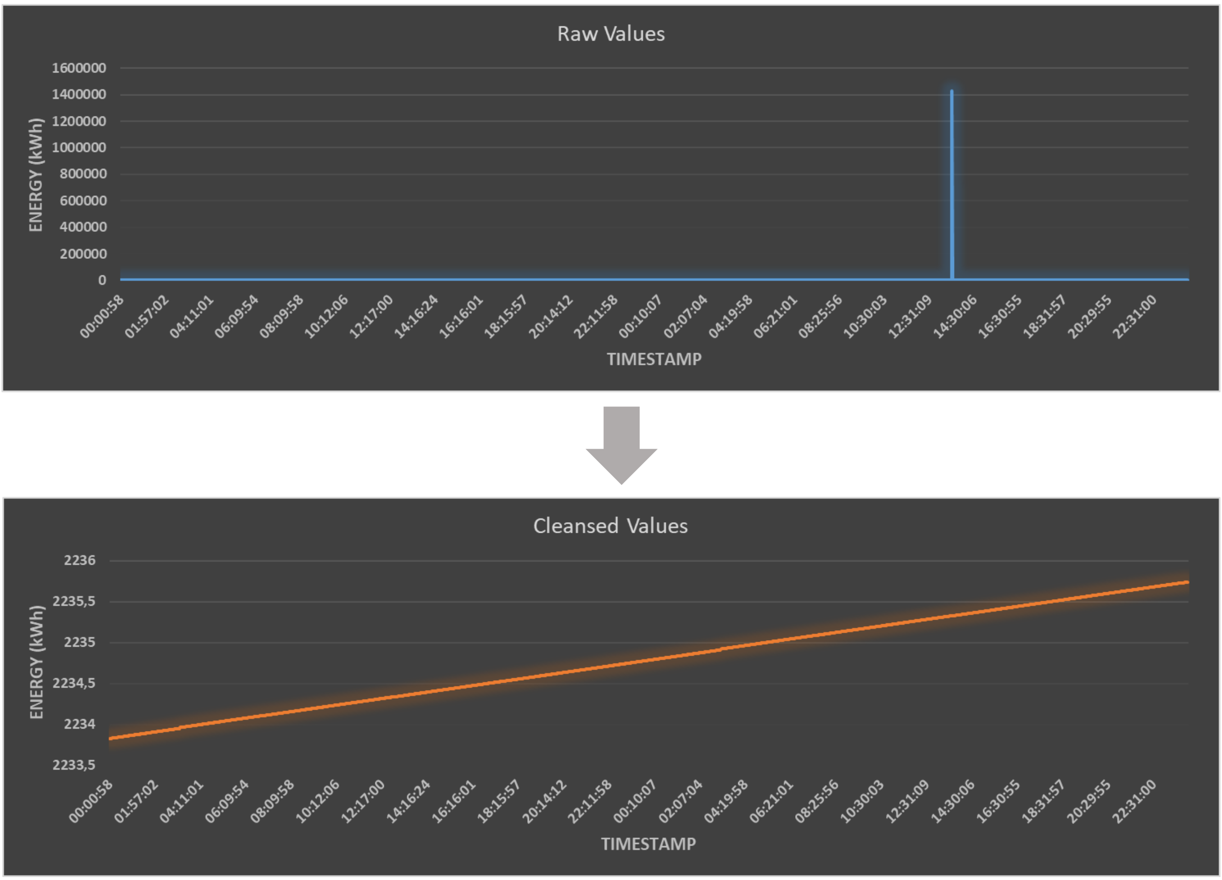


Figure 25. Hampel Filter applied on an energy consumption data set

Grubbs Test*[10]*:

The one-sided Grubb's test is a parametric hypothesis-testing method used to detect outliers in a univariate data set assumed to follow a normal distribution. The algorithm orders the data and checks whether the minimum value of the dataset is an outlier (alternative hypothesis) or not (null hypothesis). Firstly, the algorithm calculates a Grubbs statistic based on the minimum value and compares it with a critical value from a statistical table that corresponds to the assumed distribution. If the compared value is larger than the critical, the null hypothesis is rejected with predefined confidence. The algorithm iterates over all the samples in the dataset until the null hypothesis is accepted. Sequentially, the reverse process is applied for the maximum value. The Grubb's statistic used is:

$$G = \frac{|\bar{Y} - Y_{ex}|}{s}$$

Where:

$\bar{Y}$ is the sample mean

Y_{ex} is the value under examination

s is the standard deviation of the sample

a is the significance level

The Grubbs testing method recorded the lowest score in computational time among all the outlier detection techniques during the validation process. However, it offered the best results in the outlier detection in voltage values. Therefore, the Grubbs test methodology was only implemented in the voltage-time series. In Figure 26, the cleansed voltage dataset is presented in comparison with the initial raw dataset. After the removal of outliers, the voltage fluctuations are now visible.

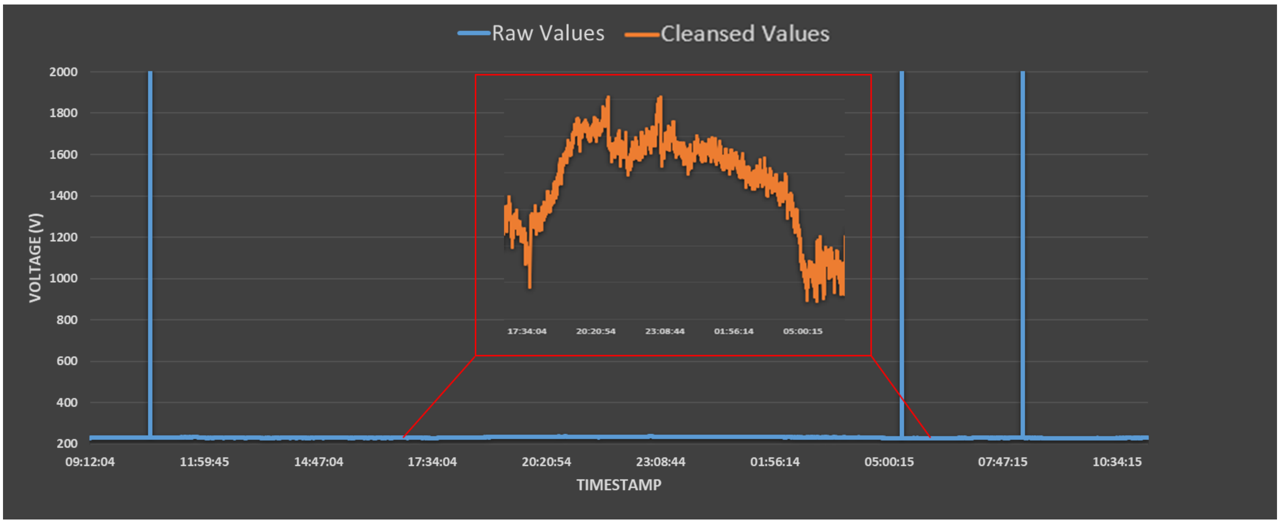


Figure 26. Grubb's test applied on a voltage data set

[10] Arshed Ahmed, Muhammad Sajjad Khan, Noor Gul, Irfan Uddin, Su Min Kim, Junsu Kim (2020), A Comparative Analysis of Different Outlier Detection Techniques in Cognitive Radio Networks with Malicious Users. Hindawi - Wireless Communications and Mobile Computing, Volume 2020, Article ID 8832191, 18 pages https://doi.org/10.1155/2020/8832191

DBSCAN[11]:

A common clustering algorithm for unsupervised learning is the density-based spatial clustering of applications with noise (DBSCAN). It is a proximity method that groups together closely packed points, while data points in low-density regions are characterized as outliers.

The algorithm accepts as input two parameters, the cluster radius 'ε' and the minimum number of samples per cluster 'minPoints'. Initiating from a random starting point, DBSCAN locates all the data points inside a radius 'ε'. Furthermore, the included data points broadcast their own perimeters (of radius 'epsilon'), leading to the addition of new members to the cluster. If no new members can be added, the cluster is finalized, and the algorithm continues the cluster formation at a new unclustered starting point until no more clusters can be formed. The values left unclustered are considered to be outlier values of the dataset. The step-by-step cluster formation process is visualized in Figure 27.

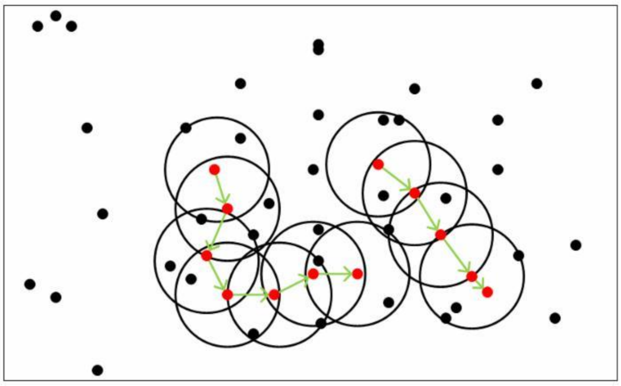


Figure 27. DBSCAN process with minPoints equal to 5[12]

Based on the lab validation results, the DBSCAN method recorded the highest score in terms of outlier detection accuracy on metrics with irregular patterns (power, luminance, and TVOC). However, a decisive disadvantage of this technique is the high computational time due to the numerous iterations in the dataset. A common solution to reduce this complexity is running the algorithm in batches, limiting the time elapsed to run. The 'batchsize' parameter defines the number of data objects that will be included in each batch. Figure 28 illustrates the DBSCAN cleansing method on a power dataset. The exclusion of the outlier discloses the real consumption peaks due to the rescaling of the y-axis.

[11] Aymen Abid, Abdennaceur Kachouri, Adel Mahfoudhi (2017), Outlier detection for wireless sensor networks using density-based clustering approach. IET Wireless Sensor Systems, doi:10.1049/iet-wss.2016.0044 www.ietdl.org

[12] ProgrammerSought, 2018, www.programmersought.com/article/68672603589/

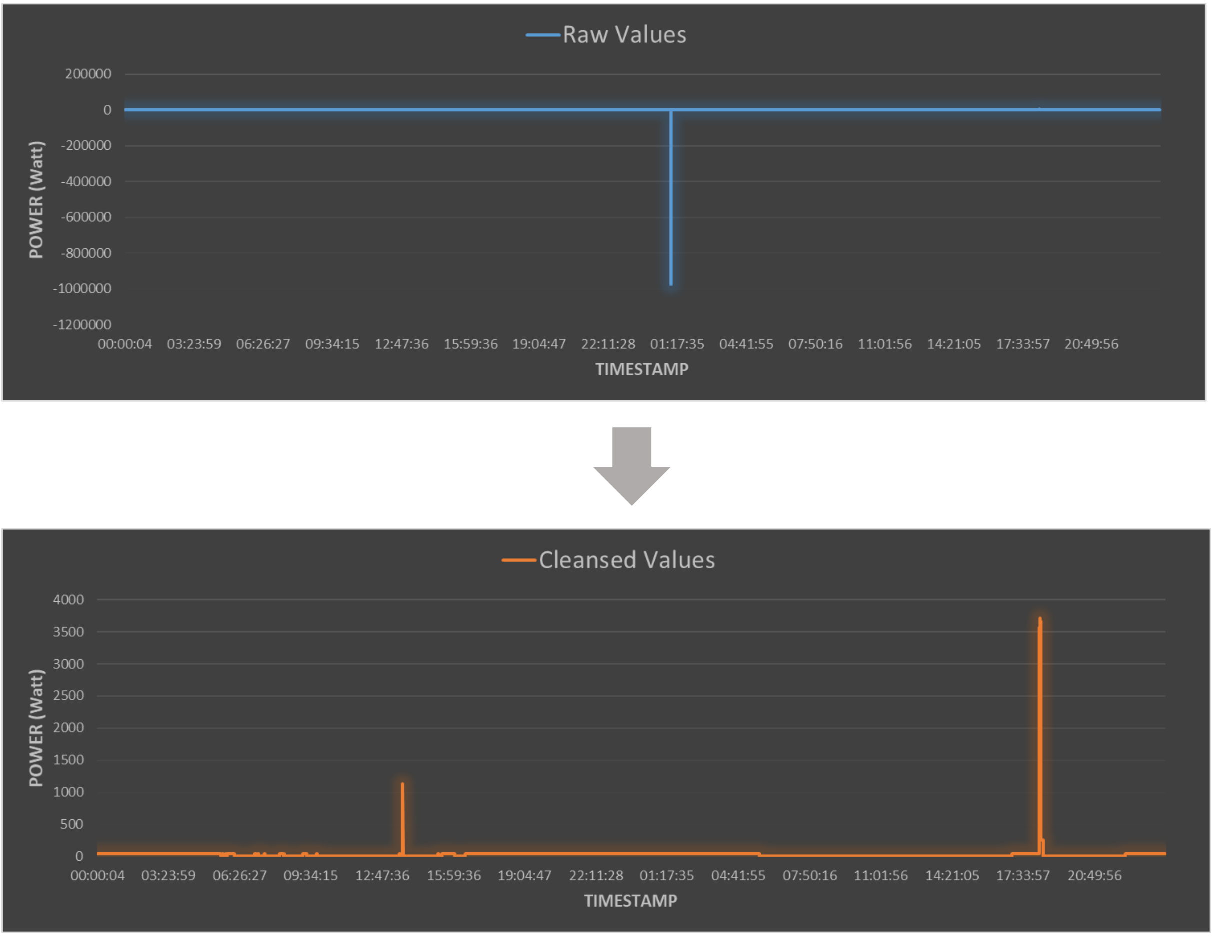


Figure 28. Power dataset before and after the DBSCAN cleansing

B.2.1.2 Outlier Treatment

The second step of the data cleansing process is the **outlier treatment**. After detecting the outlier values in the dataset, a different treatment method is utilized per metric type in order to determine whether the values should be excluded or substituted. In Table 9, a summary is presented classifying the presented outlier detection strategies per metric implemented in the cleansing tool, the corresponding treatment strategy per metric, and the fine-tuned parameters that yielded the optimal results in the lab evaluation process.

Table 9. List of outlier detection techniques per metric type.

Attribute	Outlier Detection Strategy	Outlier Treatment Strategy	Parameters
Temperature	MWFA	Mean window value	Windowsize = 8 (Two-successive filters)

Luminance	DBSCAN	Removal	Epsilon = 35, Minpoints = 20, batchsize = 100
Power	DBSCAN	Removal	Epsilon = 5000, Minpoints = 25, batchsize = 100
Voltage	Grubbs Test	Removal	a = 0,01
Current	DBSCAN	Removal	Epsilon = 25, Minpoints = 20, batchsize = 100
Energy	Hampel Filter	Median Window value	Windowsize =4 (Two-successive filters)
Humidity	MWFA	Mean window value	Windowsize = 8 (Two-successive filters)
TVOC	DBSCAN	Removal	Epsilon = 162, Minpoints = 10, batchsize = 100

As observed in Table 9, for the DBSCAN and Grubbs strategies, the outlier values are completely excluded after detection. In order to eliminate the inconsistencies, a **data imputation** methodology is applied to the cleansed data, homogeneously filling the missing values.

Apart from outlier removal, connection disruptions of the IoT devices can also lead to gaps between timestamps in the initial datasets. In order to provide a continuous output, the intermediate missing elements are filled with the last available recorded value. The same principle also applies during the outlier treatment phase when a divergent value is removed and replaced by the last valid value. The final cleansed dataset is free from deviating points and discontinuities and can be further channeled to other project components in a ready-to-process form.

B.2.2 Data Transformation

The data processing tool includes various transformation techniques to facilitate the cleansing process or transform the datasets into appropriate forms for analysis and mining. The three major categories of data transformation are the following:

- **Normalization,** where data is scaled to fall within a specified range (such as [-1,1], [0,1] etc.). Some cleansing techniques assume that the data follows a normal distribution (like Grubb's test). Hence the normalization transformation precedes the cleansing.

- **Aggregation,** where summary or grouping operations are applied to the data. Aggregation can also shift the granularity of the dataset to the desired level (from daily data to yearly etc.)
- **Generalization,** where higher-level concepts replace low-level or raw data through the use of concept hierarchies

B.3 Data Handling

As mentioned before, the Data Handling component in the IML is responsible for retrieving or communicating information to the IoT gateway. Whenever a message reaches the IML cloud from the Gateway, the handling application verifies whether it corresponds to a registered metric. If the identification fails, the message is discarded. This validation process guarantees that the message originates from a commissioned IoT device installed on the premises.

There are two types of messages coming from the Gateway

- **State-Change Events:** These messages are generated when the value of a sensor/meter (corresponding to a metric) changes by a defined percentage in comparison to the previous value or when a value is transmitted by an actuator (mode, setpoint etc.)
- **Status-Change Events:** These messages are generated when the IoT device changes connection status (online, offline, dead).

Apart from retrieving information from the Gateway, the Data Handling tool is also responsible for communicating actions to the controllers/actuators of the network. Whenever an internal or third-party application sends a request for control action, the Data Handling converts it to an appropriate format. It transfers it to the Gateway where the action is allocated to the targeted device. More specifically, initially, the tool verifies that the requested control action refers to a registered metric. Then, the Data Handling tool retrieves each unique identifier and immediately publishes a control action event including all the necessary properties to the IoT Interface component.

B.4 IoT Interface & Common Information Interfaces

The **IoT interface** subcomponent orchestrates the communication between the IoT gateway and the IML cloud in a reliable and secure manner. The module is an integration broker utilizing the Advanced Message Queuing Protocol (AMQP), which standardizes messaging using:

- **Producers** are agents that send messages
- **Queues** are buffers that store messages
- **Consumers** are agents that receive messages

The intermediary between the producer and the queue is called *exchange* and is an agent responsible for routing the messages to different queues following the *publish/subscribe* pattern. Two types of keys enable communication between the producer and the consumer. A message routing key allows the producer to send a message and a binding key, determining how an exchange connects to a queue. Two distinct types of exchanges are implemented:

- The '**direct**' exchange (Figure 29), in which the messages are sent to the queues where the routing key equals the binding key. This method is used for direct communication among the producers and consumers. The state-change and status-change events are published from the Gateway to a predefined queue and consumed by the IML cloud Data Handling tool.
- The '**topic**' exchange (Figure 30) in which the messages are sent to the queues where the routing key partially matches the binding key. This method handles the broadcast of control

actions requests to the respective IoT agents. The control actions requested from third-party applications or internal systems are published from the IML cloud Data Handling tool and received by the respective Gateway.

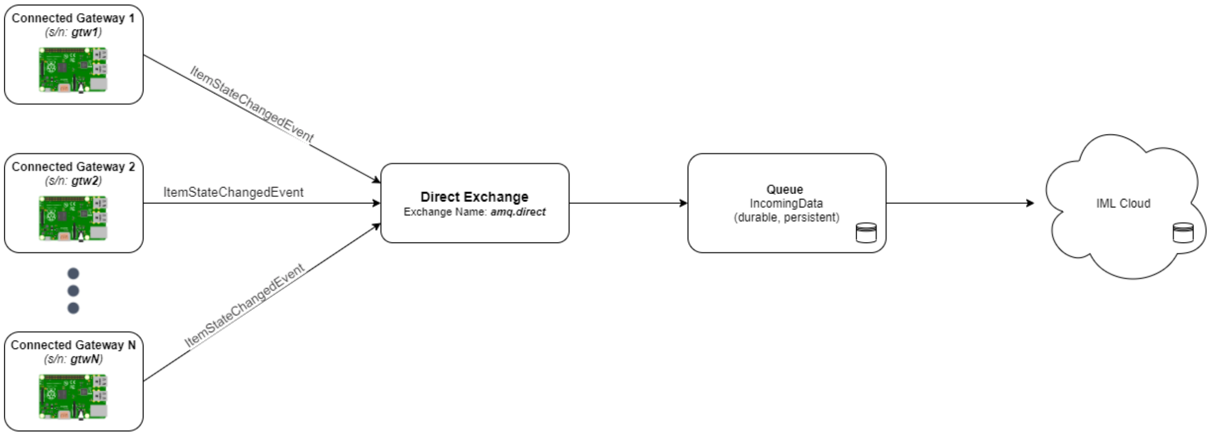


Figure 29. Direct exchange type.

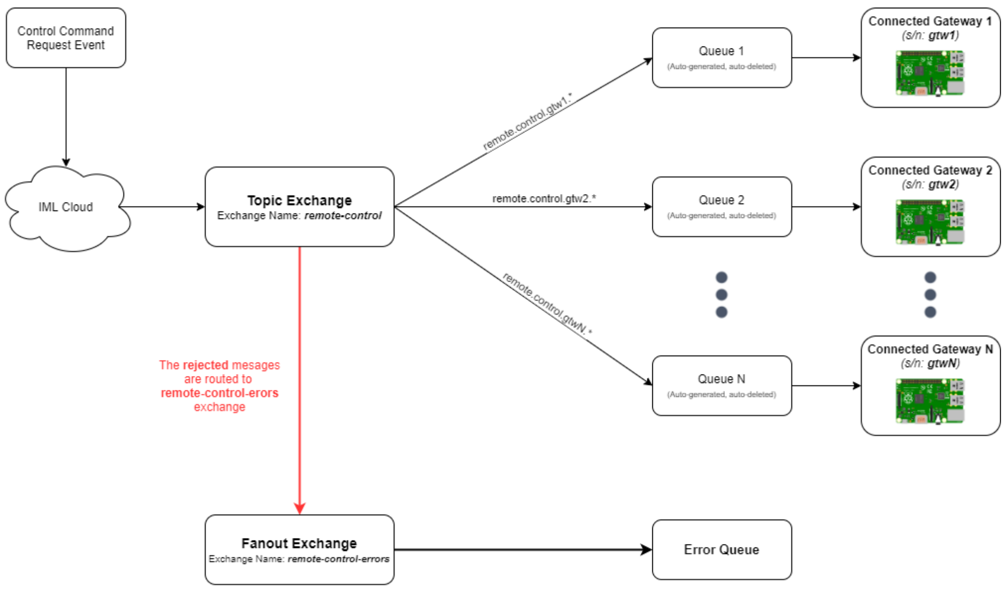


Figure 30. Topic exchange type.

Beyond the publish/subscribe pattern, another inter-process communication technique is utilized, called *Remote Procedure Call* (RPC). RPCs are commonly used with client-server architectures. The client (i.e., the Gateway) requests to act as the server (i.e. the IML cloud), and waits for its response. The messaging paradigm tries to enforce a different approach with the fire-and-forget messaging style. However, it is possible to use properly designed AMQP queues to perform and enhance RPC, as shown in Figure 31. This technique is utilized by the *Commissioning & Configuration* app, where the overall process requires separate tasks to be performed by either the Gateway or the IML cloud in a synchronous manner.

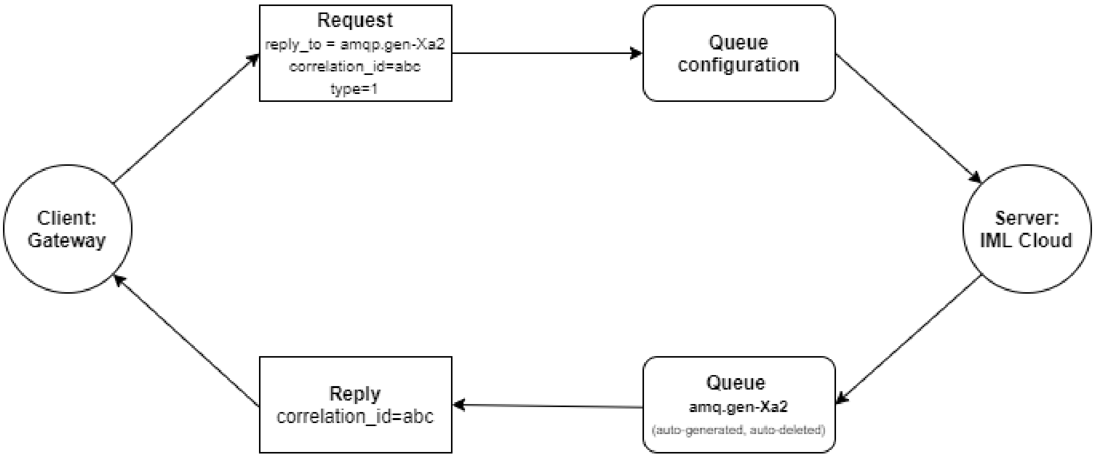


Figure 31. RPC architecture.

The **Common Information Interfaces** is the subcomponent responsible for wrapping the IML cloud data model to an external data model. This subcomponent is also responsible for the data exchange between the IML cloud and other software components. It contains two modules that regulate the user's/system permissions:

- **the Authentication** module which grants or denies access to the user/system sending a request
- **the Authorization** module determines the type of information that the user/system has access to

The communication between the cloud components is realized via a REST API under HTTPS, ensuring **encrypted** information and a more secure environment for the data exchange.

ANNEX C: Pilot Equipment Specifications

C.1 Case Study 1

C.1.1 CARLO GAVAZZI EM270

Multi-circuit power analyzer for single or threephase systems installable on panels or DIN rails. Manages current input via two current transformer blocks connected with RJ-11 connectors. The EM270 is equipped with a LCD display with controls to display measurements and configure the system, a RS485 port and two pulse outputs or two RS485 ports for daisy chain connections.

Figure 32. CARLO GAVAZZI EM270

Table 10. CARLO GAVAZZI EM270 specifications[13]

Type(connection)	3-phase/3-Wired, 3phase-4Wired or 1-phase/2-Wired
Rated Current (In)	65 A: TCD0W / 160 A: TCD1X / 250 A: TCD2X / 630 A: TCD3X
Maximum current	1.2In

[13] https://gavazziautomation.com/images/PIM/DATASHEET/ENG/EM270_DS_ENG.pdf

Minimum current	0.02 In
Starting current	0.002In
Nominal Voltage	230V L-N/400 V L-L or 120V L-N/230V L-L
Power consumption	< 4VA / 2W
Dust/water protection	IP50
Operating temperature	-25 to +55 °C
Frequency range	45 to 65 Hz
Installation	DIN 72 x 72mm
Dimensions	72 x 72 x 65 mm
Communications	Modbus RTU

C.1.2 CARLO GAVAZZI EM340

Three-phase energy meter with backlit LCD display with integrated touch keypad. Particularly indicated for active energy metering and for cost allocation in applications up to 65 A (direct connection), with dual tariff management availability. It can measure imported and exported energy or be programmed to consider only the imported one. Housing for DIN-rail mounting, with IP51 front degree protection. The meter is optionally provided with pulse output proportional to the active energy being measured, RS485 Modbus port or M-bus port. Available for legal metrology (PF option, only for imported energy).

Figure 33. CARLO GAVAZZI EM340

Table 11. CARLO GAVAZZI EM340 specifications[14]

Type(connection)	3-phase loads, direct connection
Maximum current	65
Minimum current	5
Starting current	20mA
Nominal Voltage	208 to 400V
Power consumption	≤ 1W, ≤ 10VA
Dust/water protection	IP51

[14] https://gavazziautomation.com/images/PIM/DATASHEET/ENG/EM340_DS_ENG.pdf

Operating temperature	-20 to +65 °C
Frequency range	45 to 65Hz
Installation	DIN-rail
Dimensions	54 x 90 x 63mm
Communications	Modbus RTU

C.1.3 Plugwise Sense

Sense enables wireless measuring of temperature and air humidity in a room. The temperature sensor has a range of 0 to 60˚C; the air-humidity sensor has a range of 5 to 95% RH. Sense is further capable of linking switch actions to certain measurement values

Figure 34. Plugwise Sense

Table 12. Plugwise Sense specifications[15]

Power Supply	single 3.6V AA battery
Measured temperature range	0 to 60˚C; Accuracy <±1%
Measured humidity range	5 to 95 %RH; Accuracy <±3.5%
Operational temperature	0 to 60˚C
dimensions	86 x 80 x 21 mm

C.1.4 Thermokon SR04CO2

Radio room sensor for measuring CO2 content, temperature and relative humidity (optionally) in living and office spaces. The device to be mounted via adhesive pad or screws sends its measured values unidirectionally to corresponding receivers or gateways, which process the information directly or – depending on the application – forward it to a central control unit.

Figure 35. Thermokon SR04CO2

Table 13. Thermokon SR04CO2 specifications[16]

[15] Plugwise Sense climate monitoring and management manual

[16] https://www.thermokon.de/direct/engb/products/view?path=%2F1%2F2%2F20%2F24%2F44%2F9246%2F9255

Power supply	15..24 V = (±10%)
Measuring range temp	0 to 51 °C; Accuracy ±1%
Measuring range humidity optional	0 to 100% rH (non-condensing); Accuracy ±3%
Measuring range CO2	0 to 2550 ppm; Accuracy ±75 ppm or ±10%
Operating Frequency	868 MHz
Protection	IP30
Dimensions	84.5 x 84.5 x 25 mm

C.2 Case Study 2 & 3

C.2.1 Airwits R4.1

This intelligent and small device measures and sends indoor temperature and humidity data once in every 30 minutes, and the interval can be configured remotely via the network. AirWits uses the worldwide Sigfox IoT-network connectivity for data transmission, enabling very low lifetime cost. The installation of the device is therefore extremely simple procedure, and requires neither special tools nor configuration operation. Simple, connected, maintenance free, ultra-low cost, powerful, accurate - the perfect solution for long term indoor temperature and humidity metering

Figure 36. Airwits R4.1

Table 14. Airwits R4.1 specifications[17]

Power supply	Batteries 3xAA 3.6 V LiSoCl2, 8100mAh
Measuring range temp	0 to 50 °C
Measuring range humidity	0 to 95 %
Protection	IP20
Dimensions	80 x 80 x 27 mm
Communication	Sigfox 868 MHz (RCZ1), 902 MHz (RCZ2), 920 MHz (RCZ4)

[17]https://www.connectedfinland.fi/wp-content/uploads/2017/09/AirWits_4.1_Manual_Connected_Inventions.pdf

C.2.2 Airwits CO2 R4 & R5.2

Connected AirWits CO2 is a connected carbon dioxide (CO2), temperature and humidity metering device for real and accurate indoor air quality monitoring with ultra-low lifetime costs. The device measures and sends the CO2, temperature and humidity data once in every 30 minutes, and the interval can be configured remotely via the network. AirWits CO2 uses worldwide Sigfox IoT-network for data transmission, enabling very low lifetime cost and long battery life of 5 years. Versatile, connected, maintenance free, ultra-low cost, powerful, accurate – perfect solution for easy, long term and cost-efficient indoor air quality measurements with real carbon dioxide sensor. Two AirWits CO2 models are available in the pilot premises the R4 and the R5.2.

Figure 37. Airwits CO2

Table 15. Airwits CO2 R4 specifications[18]

Power supply	3 x AA 3.6 V 12300 mAh
Measuring range temp	Temperature 0 °C to 50 °C
Measuring range humidity	Humidity 0 to 95 %
Measuring range CO2	C02 0 to 5 000 ppm

[18] Connected Airwits CO2 R4 user manual

Protection	IP20
Dimensions	100 x 100 x 27 mm
Communication	Sigfox 868 / 902 / 920 MHz

Table 16. Airwits CO2 R5.2 specifications[19]

Power supply	3xA 3.6 V LiSoCl2, 12000mAh
Measuring range temp	0 … 50 °C [accuracy 0.2°C]
Measuring range humidity	0 … 95 % [accuracy 2%]
Measuring range CO2	0…5000 ppm [typical accuracy +3%, +/-30ppm in +25°C]
Protection	IP20
Dimensions	100 x 100 x 27 mm
Communication	Sigfox 868 MHz (RCZ1), 902 MHz (RCZ2), 920 MHz (RCZ4)

[19] Connected Airwits CO2 R5.1 user manual

C.3 Case Study 4

C.3.1 HOBO® EG4115 Data Logger

The EG4115 Core is a 15-channel energy meter with 0.5% revenue-grade accuracy compliance and the ability to measure residential or commercial circuit panels, up to 3-phase 277/480VAC and 6900A. The meter has HomePlug AV powerline communication, an Ethernet port, and 2 USB ports for additional options, like WiFi. The embedded web server lets you connect to the interface over the internet or on a local area network. Each unit has a data logger that stores up to 64 data points for the lifetime of the hardware, and you can access data as granular as 1-second averages for the most recent hour of recording. The EG4115 Core is an accurate and flexible solution for monitoring multiple circuits in any residential or commercial energy monitoring application.

Figure 38. HOBO® EG4115 Data Logger

Table 17. HOBO® EG4115 Data Logger specifications[20]

Type(connection)	3-phase
Maximum current	6900A
Minimum current	5A
AC Voltage	L1: 85-277 Vrms L2: 0-277 Vrms L3: 0-277 Vrms

[20] https://www.onsetcomp.com/products/data-loggers/eg4115-core/

DC Voltage	42 Vrms
Power draw	12W max, 2W typical
Operating temperature	-30 to 70 deg C
Sampling rate	50 or 60 Hz
Dimensions	17 x 8 x 4.6cm
Communications	Ethernet: IEEE 802.3 - LAN

C.3.2 HOBO® EG4130 Data Logger

The EG4130 Pro combines an energy meter, data logger, and a web server. This powerful combination lets you measure, store and retrieve (V, A, VAr, kWh, etc.) data directly from the device or from a remote location via the internet. You can view historical and live data for the lifetime of the hardware with the unit's convenient user interface (UI). Users have access to real-time values, long-term reports, an interactive graphical interface, and many other tools

Figure 39. HOBO® EG4130 Data Logger

Table 18. HOBO® EG4130 Data Logger specifications[21]

Type(connection)	3-Phase
Maximum current	6900A
Minimum current	5A
AC Voltage	L1: 85-277 Vrms L2: 0-277 Vrms L3: 0-277 Vrms
DC Voltage	42 Vrms
Power draw	12W max, 2W typical
Operating temperature	-30 to 70 deg C
Sampling rate	50 or 60 Hz
Dimensions	17 x 8 x 4.6cm
Communications	IEEE 802.3 - LAN

C.3.3 HOBO® MX1102A Data Logger

Onset's HOBO MX1102 CO2 logger makes it more convenient than ever to measure and record CO2 in buildings and other noncondensing environments. It measures CO2 from 0–5,000 parts per million (ppm) – and our free HOBOconnect app lets you access data right from your mobile phone, tablet, or

[21] https://www.onsetcomp.com/products/data-loggers/eg4130-pro/

Windows computer when within a 100-foot range of the logger. The MX1102 also features a USB port so it can be used with a computer running HOBOware graphing & analysis software.

Figure 40. HOBO® MX1102A Data Logger

Table 19. HOBO® MX1102A Data Logger specifications[22]

Power supply	4 AA 1.5 Volt batteries (user replaceable) or USB power source (5 V DC, 2 Watts)
Measuring range temp	0° to 50°C
Measuring range humidity	1% to 90% RH
Measuring range CO2	0 to 5,000 ppm
Protection	IP50
Dimensions	7.62 x 12.95 x 4.78 cm
Communication	Bluetooth

[22] https://www.onsetcomp.com/products/data-loggers/mx1102a/