WP4 - Development of an interoperable and secure PHOENIX Smartness Hub based on an ICT solution

Document Version:

D4.1 PHOENIX Smartness Hub implementation - Initial Version

2.0

Project Number:	**Project Acronym:**	**Project Title:**
893079	PHOENIX	Adapt-&-**P**lay Holistic c**O**st-**E**ffective and user-frieNdly **I**nnovations with high replicability to upgrade smartness of e**X**isting buildings with legacy equipment

Contractual Delivery Date:	**Actual Delivery Date:**	**Deliverable Type* - Security**:**
30/06/2021	30/06/2020	R - PU

* Type:	P - Prototype, R - Report, D - Demonstrator, O - Other
** Security Class:	PU- Public, PP - Restricted to other programme participants (including the Commission), RE - Restricted to a group defined by the consortium (including the Commission), CO - Confidential, only for members of the consortium (including the Commission)

Responsible and Editor/Author:	**Organization:**	**Contributing WP:**
Josiane Xavier Parreira	SAGOE	WP4

Authors (organizations):

SAGOE, UMU, OdinS, UBITECH

Abstract:

This document details the initial version of the Smartness Hub implementation. The current implementation has been focused on the PoC, while the next versions will be extended to the other pilots. The document describes the Knowledge Graph at the edge level, accomplished by NGSI-LD compliant IoT brokers, the different building models being considered, namely, SAREF, BRICK and Smart Data Models, as well as a model for the SRI assessment, developed within the project. The document also describes the current status of the user and grid services being developed. Current services access data directly from NGSI-LD brokers, while future services, such as the SRI computation, will make use of data analytics tools hosted in a dedicated graph storage. Finally, the document also includes the current status of the security and privacy features being developed for the PoC.

Keywords:

Smartness Hub; Knowledge Graph; NGSI-LD, SRI, Security, Privacy.

Disclaimer: The present report reflects only the authors’ view. The European Commission is not responsible for any use that may be made of the information it contains.

Revision History

The following table describes the main changes done in the document since created.

Revision	Date	Description	Author (Organization)
0.1	26/05/2021	ToC draft	Josiane Parreira (SAGOE)
0.2	09/06/2021	User services	Dimitra Georgakaki (UBITECH)
0.3	14/06/2021	SRI model and computation	Stefan Bischof (SAGOE)
0.4	16/06/2021	Building data models	Josiane Parreira (SAGOE)
0.5	18/06/21	Integration of PoC description, semantic models and grid integration services	Alfonso Ramallo, Pedro González Gil, Aurora Gonzalez Vidal (UMU)
0.6	18/06/21	Privacy and security section	Rafael Marín Pérez (OdinS)
0.7	23/06/21	Document merge	Josiane Parreira (SAGOE)
1.0	28/06/21	Reviews' integration	Josiane Parreira (SAGOE)
1.1	29/06/21	Authors' revisions	All authors
2.0	30/06/21	Minor formatting changes	SAGOE/UMU

Executive Summary

This document documents the work in WP4 towards the development and implementation of PHOENIX Smartness Hub. As initially planned, the current implementation has been focused on the Proof of Concept (PoC) to allow fast prototyping, while the next versions will be extended to the other pilots.

The document describes the first instance of the PHOENIX Smartness Hub, which was done at the PoC site in Murcia, Spain, its appliances, and the sensors that have been deployed. Communication with the sensors is enabled by different IoT connectors, such as MQTT brokers. Both sensors and their data are then semantically annotated. For that, PHOENIX combines and extends three semantic models geared to IoT devices: Smart Data Models, IoT-Stream, and QUDT, which are described in this document. The semantic representations of sensors and sensor data are then stored in NGSI-LD complaint brokers. This combination of semantic data at IoT brokers is PHOENIX's realization of Knowledge Graph at edge level. The edge KG's can be directly query via NGSI-LD interfaces available at the brokers. Besides information about sensor and sensor data, other building properties are relevant to the project, such as its structure, topology, the appliances installed and their location within the building, etc. While it is possible to store such information at the edge KG, PHOENIX resorts to a cloud-based KG based on graph storage, to order to represent higher level data and building's properties in a more efficient manner. The cloud KG will make use of existing building models such as SAREF, BRICK and BOT to represent the buildings. These models are briefly introduced in this deliverable. It will also subscribe to the edge KGs to retrieved sensor descriptions. The KGs (both edge and cloud) will support the different analytical services within PHOENIX. One particular service is the Smart Readiness Indicator (SRI) assessment and SRI score computation. PHOENIX will develop methods to provide a semi-automatic building assessment and automatic SRI score computation. To that end, we have developed a semantic model to represent SRI related features, such as the buildings' assessment, the SRI domain/impact hierarchy, as well as the SRI scores and weight matrices. This deliverable describes the SRI model and how it can be used to compute the SRI scores.

The KG is the core of the AI-based Knowledge Engine, which is part of the Knowledge Layer. The remaining two components within the Knowledge Layer are the User-centric services Analytics Engine and the Grid-centric services Analytics Engine. As the names suggest, the User-centric engine will support the user-centric services in the Function layer, whereas the Grid-centric

engine will deliver algorithms to be used in the grid-oriented services.

At the initial stage the user-centric services are geared to comfort optimization. The user-centric engine will devise comfort optimization algorithms which consider multiple data inputs, both users' direct actions (e.g., profile and active temperate settings) and passive data coming from sensors installed in the building. The user experience will also be enhanced by delivering insightful forecasts to the users, for better awareness and assurance of optimal indoor conditions. The grid-centric engine is currently developing and testing different AI algorithms for energy availability forecasting. As grids are rather complex and often difficult to monitor, we have created a synthetic micro-grid for better deployment and testing of services. This micro-grid realistically simulates human occupancy and behaviour as well as buildings' appliances. In addition to the sensor data, the meters, the weather information, and the framework for the synthetic simulation of the grid, the services for grid integration, the grid engine will also make use of real data coming from REE, a TSO of Spain, which also operates in other countries.

Finally, the document also details the security & privacy framework implemented within the project. In particular, we describe the risk analysis performed at the early stage of the project, which lead to a series of features to be implemented. Afterwards, we provide a review of the identified features and how they could be instantiated within the PHOENIX Smartness Hub. Finishing up with the implementation of security that has been carried out in the Gateways of Z-wave of the PoC.

Disclaimer

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 893079, but this document only reflects the consortium's view. The European Commission is not responsible for any use that may be made of the information it contains

Table of Contents

Table of Figures

Acronyms

Abbreviation	**Description**
AA	Attribute Authority
ABAC	Attribute-based access control
ABE	Attributed-Based Encryption
AHU	Air Handling Unit
API	Application Programming Interface
ARIMA	AutoRegressive Integrated Moving Average
BIM	Building Information Modelling
BMS	Building Management System
BOT	Building Topology Ontology
CapBAC	Distributed Capabilities-Based Access Control
ECC	Elliptic Curve Cryptography
ESCO	Energy Service Company
ETSI	European Telecommunications Standards Institute
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
HVAC	Heating, ventilation, and air conditioning
IBE	Identity-Based Encryption
ICT	Information and communications technology
IdM	Identity Management
IFC	Industry Foundation Classes
IoT	Internet of Things
JSON	JavaScript Object Notation
KG	Knowledge Graph
LSTM	Long short-term memory
MQTT	Message Queuing Telemetry Transport
NGSI	Next Generation Service Interfaces
NGSI-LD	Next Generation Service Interfaces – Linked Data
OWL	Web Ontology Language
PAP	Policy Administration Point
PEP	Policy Enforcement Point
PoC	Proof of Concept
QB	W3C RDF Data Cube ontology

QUDT	Quantities, Units, Dimensions, and Types Ontology
RBAC	Role-based access control
RDF	Resource Description Framework
REE	Red Eléctrica de España
REST	Representational state transfer
S&P	Security & Privacy
SAML	Security Assertion Markup Language
SAREF	Smart Applications REFerence
SAREF4BLDG	SAREF extension for building
SCADA	Supervisory control and data acquisition
SOSA	Sensor, Observation, Sample, and Actuator Ontology
SRI	Smart Readiness Indicator
SSL	Secure Sockets Layer
STF	Specialist Task Force
TSO	Transmission System Operator
URI	Uniform Resource Identifier
VRF	Variable Refrigerant Flow
W3C	World Wide Web Consortium
XACML	eXtensible Access Control Markup Language
XML	Extensible Markup Language

1 Introduction

At the central of the innovation created in PHOENIX there is a central intelligence that has been given the name of PHOENIX Smartness Hub. The Hub is a key component within the PHOENIX platform, and it will be equipped with the advanced ICT features to make sure that robustness plasticity and functionality are maximised, always with the aim of improving energy efficiency, and the improvement of services for the users that will maximise comfort, productivity, and health. To this end, the Smartness Hub makes use of semantic data models that are necessary for providing interoperability between the different families of devices, therefore presenting a "unified" system. Providing a unified representation by no means results in a centralised approach. In fact, the PHOENIX Smartness Hub is combination of edge and cloud storages, thus enabling a scalable, decentralized solution. A federation of edge devices will implement a distribute edge Knowledge Graph, which will hold information about devices and their data. These will in turn will interact with the cloud storage to provide data and data insights form sensors, for a higher-level representation and connection with building models and other data sources. The edge-cloud data storage is the backbone for the development of multiple analytical engines which are geared towards building occupants and the grid. Moreover, as the Smartness Hub acts as the middleware between data sources and services/applications, it is key that the security and privacy requirements from data and data owners are guaranteed. The Smartness Hub implements a Security & Privacy framework which handles secure authentication, access controls, identify management, and group sharing policies.

The development of the PHOENIX Smartness Hub is carried out within WP4, and its progress will be reported at three different stages of the development. This document describes the initial version of the PHOENIX Smartness Hub. As planned, the current implementation has been focused on the Proof of Concept (PoC) to allow fast prototyping, while the next versions will be extended to the other pilots. The document describes the PoC site in Murcia, Spain, its appliances, and the sensors that have been deployed. Communication with the sensors is enabled by different IoT connectors, such as MQTT brokers. For the data semantic modelling of sensors and sensors' data, PHOENIX combines and extends three semantic models geared to IoT devices: Smart Data Models, IoT-Stream, and QUDT, which are described in this document. The semantic representations of sensors and sensor data are then stored in NGIS-LD complaint brokers. This combination of semantic data at IoT brokers is PHOENIX's realization of Knowledge Graph at edge level. The edge KG's can be directly query via NGSI-LD interfaces available at the brokers. Besides information about sensor and sensor data, other building properties are relevant to the project, such as its structure, topology, the appliances installed and their location within the building, etc. The cloud KG will make use of existing building models such as SAREF, BRICK and BOT to represent the buildings. These models are introduced in this deliverable. The KGs (both edge and cloud) will support the different analytical services within PHOENIX. One particular service is the Smart Readiness Indicator (SRI) assessment and SRI score computation. PHOENIX is currently developing methods to provide a semi-automatic building assessment and automatic SRI score computation. In this document we describe the semantic model developed to represent SRI related features, such as the buildings' assessment, the SRI domain/impact hierarchy, as well as the SRI scores and weight matrices. Moreover, we also describe the SRI model and how it can be used to compute the SRI scores.

Within the PHOENIX architecture, the KGs are the core of the AI-based Knowledge Engine, which is part of the Knowledge Layer. The remaining two components within the Knowledge Layer are the User-centric services Analytics Engine and the Grid-centric services Analytics Engine. As the names suggest, the User-centric engine will support the user-centric services in the Function layer, whereas the Grid-centric engine will deliver algorithms to be used in the grid-oriented services.

At the initial stage the user-centric services are geared to comfort optimization. This document describes the user-centric engine that will devise comfort optimization algorithms which consider multiple data inputs, both users' direct actions and passive data coming from sensors installed in the building. The user experience will also be enhanced by delivering insightful forecasts to the users, for better awareness and assurance of optimal indoor conditions. The grid-centric engine is currently developing and testing different AI algorithms for energy availability forecasting. As grids are rather complex and often difficult to monitor, we have created a synthetic micro-grid for better deployment and testing of services. This micro-grid realistically simulates human occupancy and behaviour as well as buildings' appliances. In addition to the sensor data, the meters, the weather information, and the framework for the synthetic simulation of the grid, the services for grid integration, the grid engine will also make use of real data coming from REE, a TSO of Spain, which also operates in other countries.

Finally, the document also details the security & privacy framework implemented within the project. In particular, we describe the risk analysis performed at the early stage of the project, which lead to a series of features to be implemented. Afterwards, we provide a review of the identified features and how they are instantiated within the PHOENIX Smartness Hub. More specifically, the security & privacy framework currently includes components for privacy-preserving identity management, and distributed access control.

The remainder of this document is structured as follows: Section 2 describes the PoC site, the devices and the data available, where Section 3 describes the how edge KG are accomplished within PHOENIX. For the cloud KG, PHOENIX will make use of different building models, which are described in Section 4. Moreover, Section 4 also presents the semantic model and framework for supporting SRI building assessments and automatic SRI score computation. Section 5 covers the data analytics services, both user and grid centric, while the security & privacy framework is described in Section 6. Section 7 concludes the document.

2 PoC description

The first instance of the Smartness Hub, on a functional level, has been on the Proof of Concept (PoC). Here, the full path of knowledge generation has been created: from the lowest level, where the sensors are reporting data, to the top levels, where the knowledge is created. At this stage the algorithms tested are rather simple, and they only make basic calculations to obtain simple information such as occupation, comfort, or energy consumption, but they serve their purpose as they proof that the connections between the different components of the platform of the PoC are working adequately.

The PoC has been deployed in a sub-set of buildings of the University of Murcia, in the Campus of Espinardo, and it involves multiple sub systems, such as charging stations and HVAC systems. The main aim is to create a platform where all devices - both those already integrated into the existing Building Management System (BMS) and those essentially isolated - can connect and communicate. To create such communication, several interventions had to be done on the building at several levels. The interventions have been chosen according to a feasibility study of the services, in which the impact on the SRI and the interest in the PHOENIX framework were evaluated, also taking into account which ones were the most cost-effective solutions. A brief description of the main intervention is presented in this section. More detailed information can be found on Deliverable 3.1 – "Technical Upgrades and integration mechanisms for legacy equipment" and Deliverable 7.1 – "First feedback from the Proof of Concept deployment and Introduction to the other pilots".

Starting from the domestic hot water installation, this system was composed by a dual tank system, one storing hot water heated up using solar energy and one using a thermal resistance. The monitoring is increased by measuring the temperature entering the primary solar circuit, and leaving the solar circuit. Moreover, a water consumption meter was added, that works both as a global counter and a flow meter. Hence, communication is increased among the tanks, the temperature probes, and the water meter. Thanks to an IoT connector, the data is then integrated in PHOENIX via an MQTT IoT-Agent. Figure 1 shows a few pictures of the deployment.

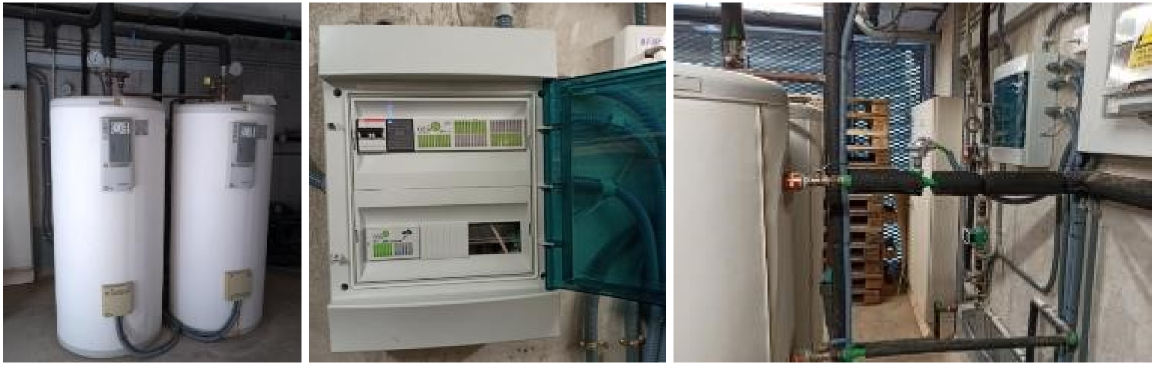


Figure 1 - Photographs of the installations performed.

Regarding the electric vehicles' charging point circuit, a second electric control panel was added to the initial installation (consisting of one circuit with two charging points), and power meters and controllers have been installed. After this intervention, scheduling can be restricted in order to charge the vehicles when the electricity is cheaper or greener (or any other criteria), and other actions such as remote actuation is now possible. Hence, data reading, actuation and integration with PHOENIX MQTT IoT-Agent have been enabled. Figure 2 depicts the installation on the charging point.

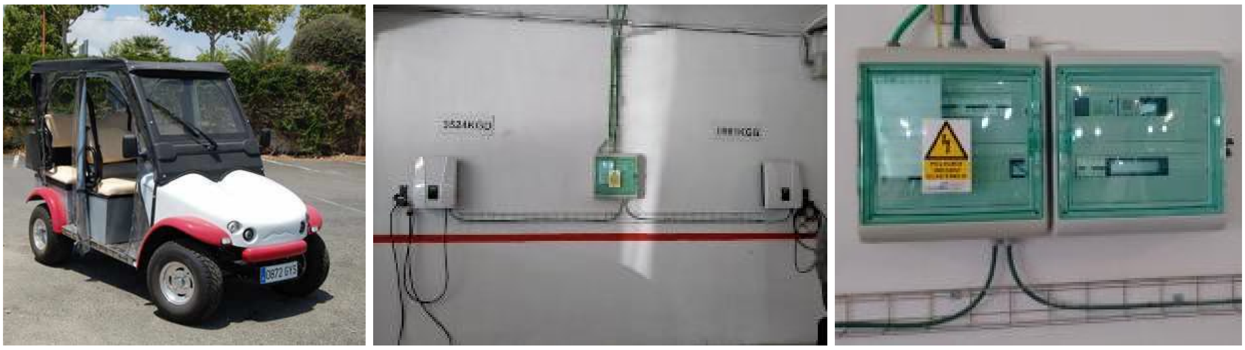


Figure 2 - Installation of meters and controllers for the electric cars' charging point.

The HVAC system is managed through a centralized VRF system with individual consoles in the internal zones. Moreover, the building has an air handling unit (AHU) with heat recovery. The measuring and actuation capacity consisted of the insertion of power meters in three different circuits: one for the AHU and two for the conditioning. Also in this case, the data reading and the integration with PHOENIX MQTT IoT-Agent were enabled.

In the PoC, special attention has been dedicated to air quality monitoring. Several CO_2 sensors have been installed and connected to the platform. The connection was enabled through two

Raspberry-Pi with Z-Wave adapters (Figure 3). Also, the sensors sending data to a SCADA system have been made accessible via middleware. The solution implies a great improvement for the comfort and wellbeing of occupants, which showed particular interest in the topic considering the recent pandemic.

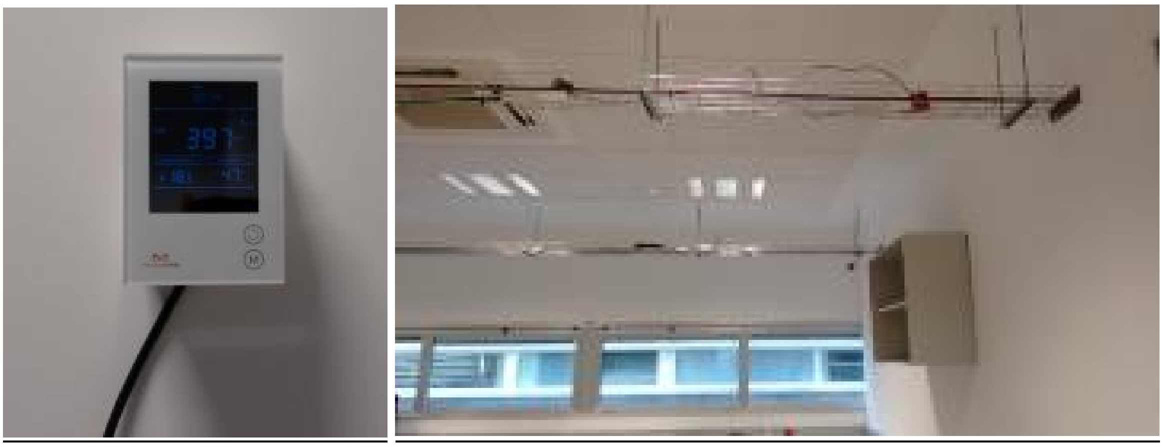


Figure 3 - CO2 monitor (left) and RaspberryPi-mounted connectivity system (right).

To improve energy efficiency and smartness, consumption's control is a key parameter. That is why the insertion of WiFi Smart Sockets installed on devices has been a big step forward in the analysis of the consumption behaviour of the building's users. A Tasmota firmware[1] was configured to connect these devices to the PHOENIX platform.

A photovoltaic solar installation has been also connected to the PHOENIX platform via FTP connection. A middleware component was developed to enable access to the FTP Server and send this information to the PHOENIX hub using an MQTT/SSL interface.

One further step that improved connectivity while increasing the smartness of the building was the integration of external data sources of weather, in particular from Weatherbit[2]. Hence, real-time data and forecast weather condition are both available on the PoC's site. The Weatherbit API has a free plan for 125 API requests per day that works with just latitude and longitudes coordinates. The information is then sent to the PHOENIX hub through the MQTT/SSL interface.

[1] *https://github.com/arendst/Tasmota*

[2] *https://www.weatherbit.io/*

3 Knowledge graph at edge level - NGSI-LD and Smart Data Models

The platform created has allowed to implement the data entities that will represent the data providing devices of the real world on the PHOENIX solution. The current architecture is depicted in Figure 4. At this first stage, the PoC focuses on enabling the knowledge graph at the edge level. For this effort, the Orion Context Broker[3] has been used, that deals with the requests of the different components. The Context Broker stores the definition of the entities, which will later be used to derive higher level information that will be linked to the building models stored at the cloud level Knowledge Graph.

For representing entities at the edge knowledge graph, PHOENIX combines and extends different data models. Some of these are part of the FIWARE data-models called *"smart data models"*. Smart Data Models[4] is a collaborative initiative, supported by the FIWARE foundation[5], to provide open-license data models for different domains, such as Smart Cities, Smart Agrifood, Smart Environment, Smart Energy and Smart Manufacturing. The main characteristic of these models is their compliance with FIWARE NGSI version 2 and NGSI-LD specification, and therefore suitable for describing data within FIWARE context brokers. For representing the stream of data coming from the sensors, we use concepts from the IoT-Stream ontology[6], developed within the IoTCrawler project[7], which in turn extends the W3C's SOSA ontology[8] to describe sensor observations. Finally, the QUDT vocabularies[9] are used to represent the measurements' units. Figure 5 provides an example of the representation of a CO_2 device located within a building's zone. The devices' descriptions are then mapped to the NGSI-LD interface and stored at the context brokers. Figure 6 and Figure 7 show an example of a query to a context broker and its output, respectively.

[3] *https://fiware-orion.readthedocs.io/en/master/*

[4] *https://smartdatamodels.org/*

[5] *https://www.fiware.org/*

[6] *http://iot.ee.surrey.ac.uk/iot-crawler/ontology/iot-stream/#IotStream*

[7] *https://iotcrawler.eu/*

[8] *https://www.w3.org/TR/vocab-ssn/*

[9] *https://www.qudt.org/*

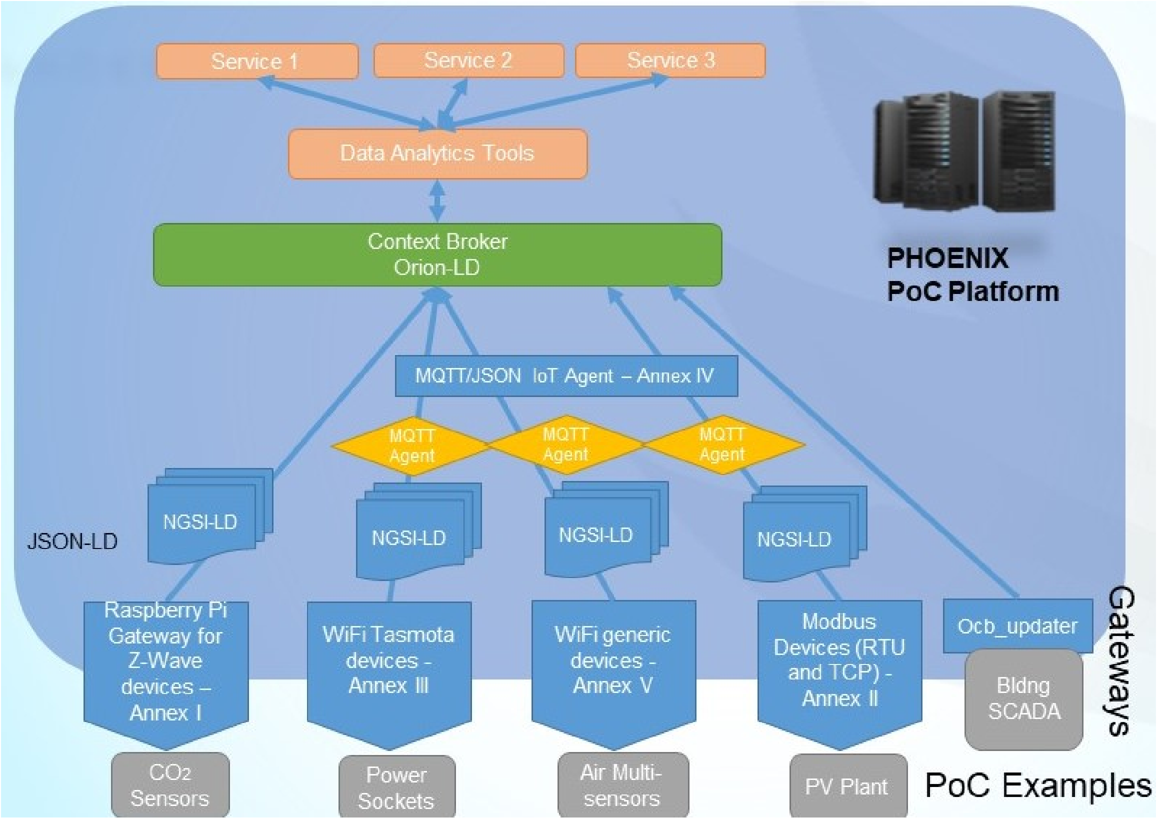


Figure 4 - Representation of the platform used for the PoC with an emphasis on the connection mechanisms of the legacy equipment.

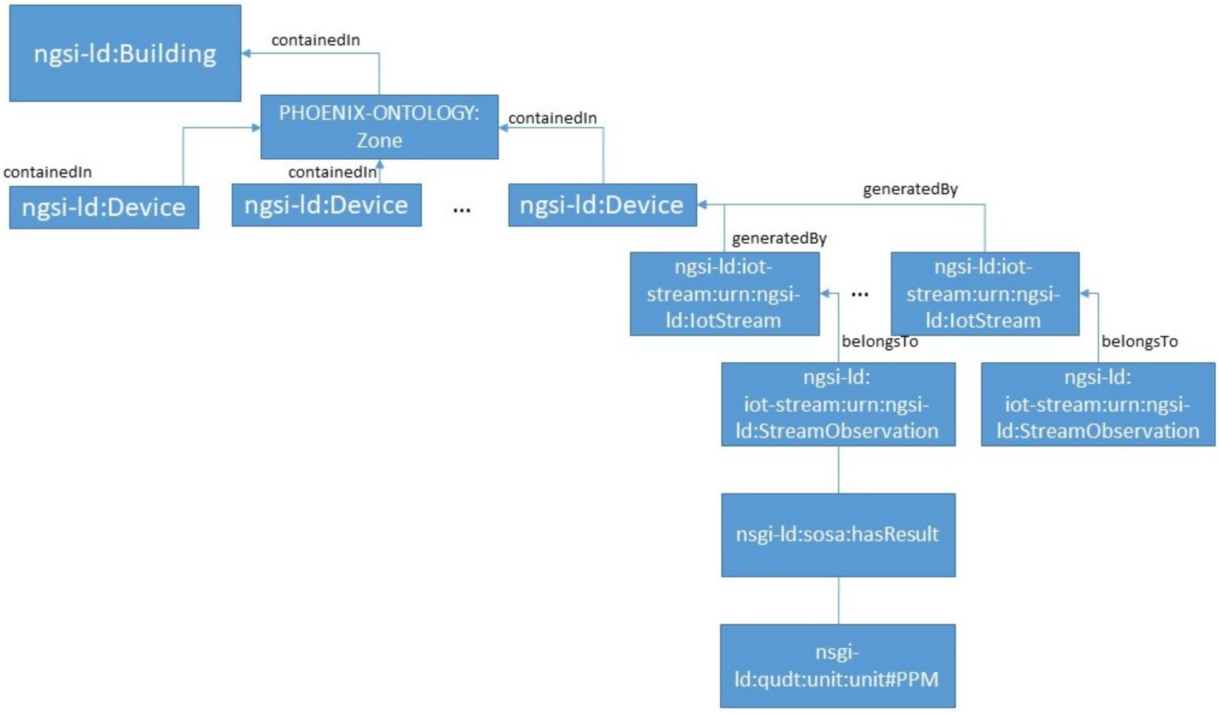
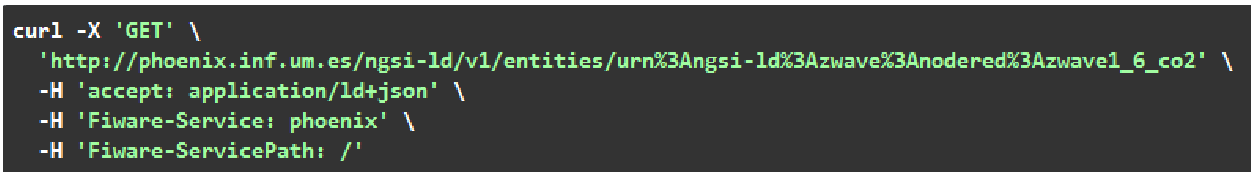
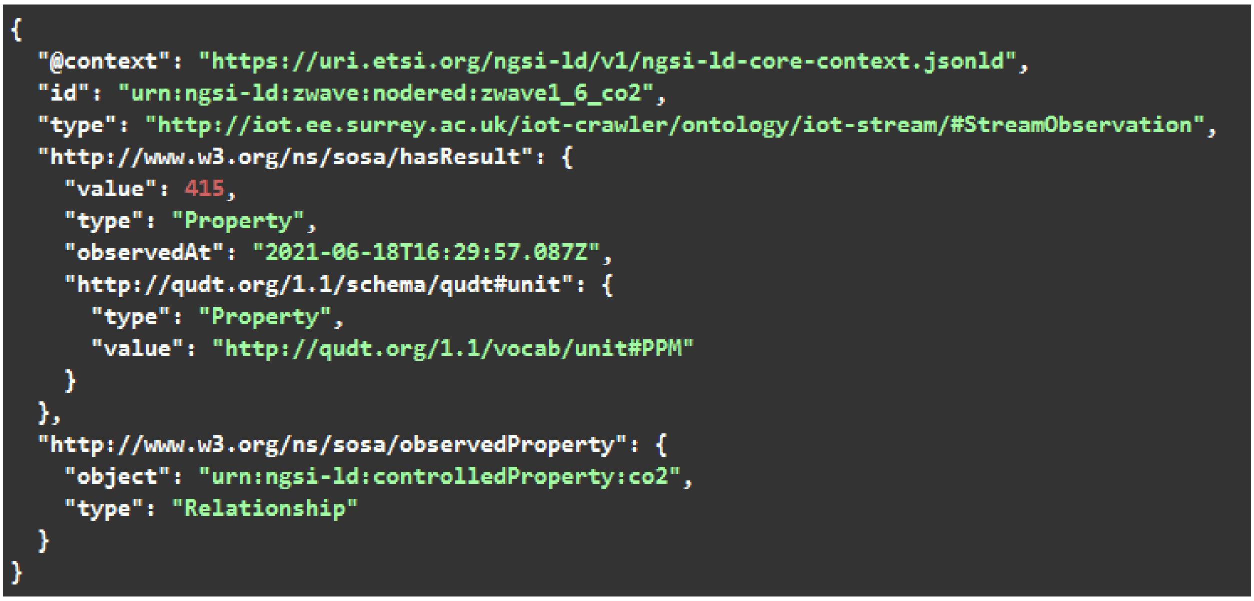

ngsi-ld:Building
containedIn
PHOENIX-ONTOLOGY: Zone
containedIn
containedIn
containedIn
ngsi-ld:Device
ngsi-ld:Device
...
ngsi-ld:Device
generatedBy
generatedBy
ngsi-ld:iot-stream:urn:ngsi-ld:IotStream
...
ngsi-ld:iot-stream:urn:ngsi-ld:IotStream
belongsTo
belongsTo
ngsi-ld: iot-stream:urn:ngsi-ld:StreamObservation
ngsi-ld: iot-stream:urn:ngsi-ld:StreamObservation
nsgi-ld:sosa:hasResult
nsgi-ld:qudt:unit:unit#PPM

Figure 5 - Diagrammatic view of the classes and relationships used for the Context Broker on an entity of a CO_2 sensor.



```
curl -X 'GET' \
  'http://phoenix.inf.um.es/ngsi-ld/v1/entities/urn%3Angsi-ld%3Azwave%3Anodered%3Azwave1_6_co2' \
  -H 'accept: application/ld+json' \
  -H 'Fiware-Service: phoenix' \
  -H 'Fiware-ServicePath: /'
```

Figure 6 - Context Broker information query.



```
{
  "@context": "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld",
  "id": "urn:ngsi-ld:zwave:nodered:zwave1_6_co2",
  "type": "http://iot.ee.surrey.ac.uk/iot-crawler/ontology/iot-stream/#StreamObservation",
  "http://www.w3.org/ns/sosa/hasResult": {
    "value": 415,
    "type": "Property",
    "observedAt": "2021-06-18T16:29:57.087Z",
    "http://qudt.org/1.1/schema/qudt#unit": {
      "type": "Property",
      "value": "http://qudt.org/1.1/vocab/unit#PPM"
    }
  },
  "http://www.w3.org/ns/sosa/observedProperty": {
    "object": "urn:ngsi-ld:controlledProperty:co2",
    "type": "Relationship"
  }
}
```

Figure 7 - Output of context request at the real time value.

As mentioned before, the data from the edge devices will be linked to higher level description of the buildings and their appliances. This work is currently ongoing. In the next section we provide a brief description of the building models that are under consideration. Moreover, we describe the SRI model, developed within PHOENIX. The SRI model extends buildings' models to allow the semantic representation of SRI assessments and the automatic SRI score calculation, which is one of the key services within PHOENIX.

4 Building data models

4.1 Overview of existing building data models

There has been multiple, parallel work on providing semantic models to describe buildings and their characteristics, such as building's layout, its appliances and their operators and devices/actuators available within a building. Furthermore, semantic models for other relevant concepts, such as geographic location and users' profile, are also available. PHOENIX envisions the use of a combination and extension of existing models in order to fulfil the services' requirements within the project. We have identified three models that are very fitting to the project, namely SAREF (and its extension SAREF4BLDG), BRICK and the Building Topology Ontology (BOT). They will be selected depending on both the data available at each pilot and the requirements for the pilot's services. Next, we provide a short description of each of these models.

The Smart Applications REFerence (SAREF) ontology[10] is a shared model of consensus that facilitates the matching of existing assets in the smart applications domain. SAREF is being developed as part of two Specialist Task Forces (STFs) within ETSI. Besides the main model, SAREF has been extended to specific domains such as Energy, Smart Cities, Water, Industry, and Agriculture. In particular, SAREF4BLDG[11] is SAREF's extension for the Smart Building domain. Figure 8 shows the top levels of the SAREF4BLDG ontology.

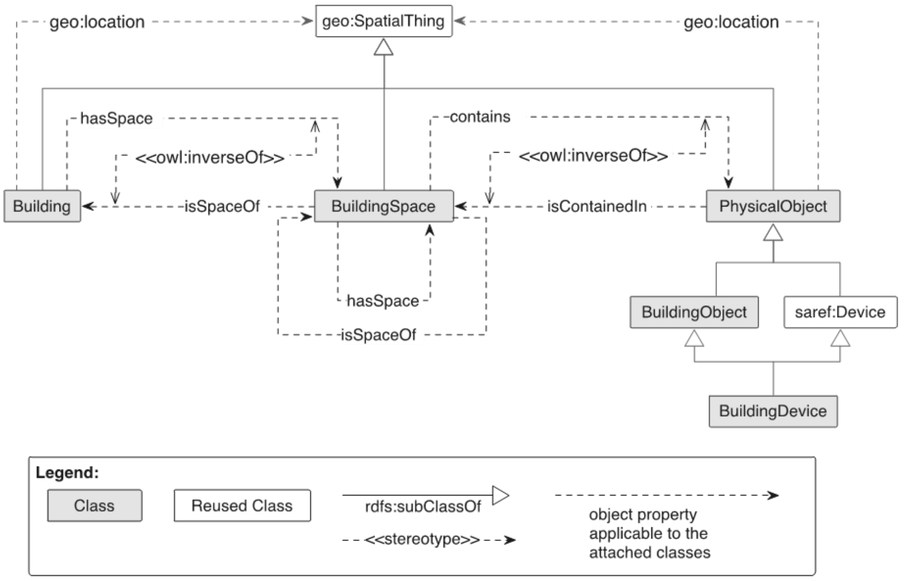


Figure 8 - General overview of the top levels of the SAREF4BLDG extension.

[10] *https://saref.etsi.org/*

[11] *https://saref.etsi.org/saref4bldg*

BRICK Schema[12] is another model that provides semantic descriptions of the physical, logical and virtual assets in buildings and the relationships between them. BRICK has only a few main concepts and properties (see Figure 9). In addition, different class hierarchies are provided in order to describe more specific concepts. A few class hierarchies' examples are shown in Figure 10.

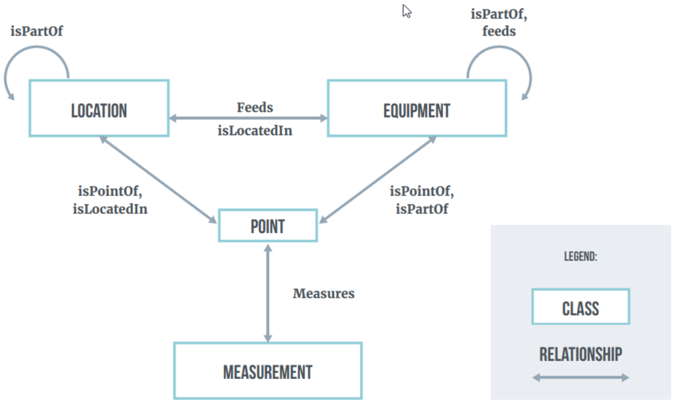


Figure 9 - BRICK main classes and relationships[13].

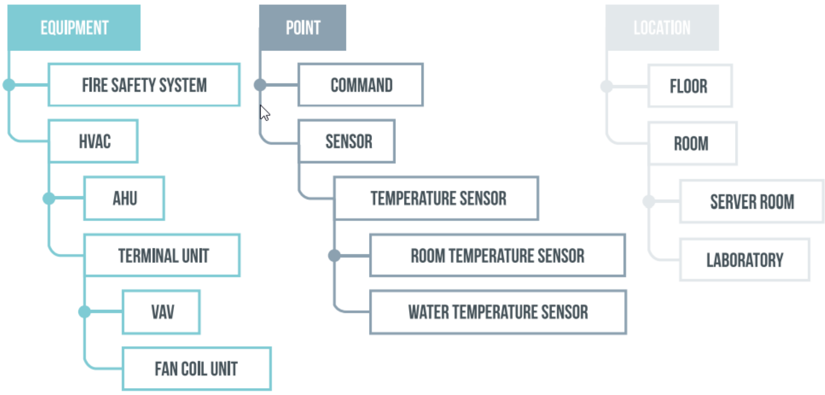


Figure 10 - Example of BRICK's class hieararchies[13].

While SAREF and BRICK focus on sensors and appliances within buildings, the Building Topology Ontology (BOT)[14] focuses, as the name suggests, on describing the building's topology. BOT is one of the outcomes of the W3C's Linked Building Data Community Group. The group also provide a tool to convert BIM models in IFC format to RDF graphs following their models.

[12] *https://brickschema.org/*

[13] *https://www.memoori.com/wp-content/uploads/2016/06/Brick_Schema_Whitepaper.pdf*

[14] *https://w3c-lbd-cg.github.io/bot/*

4.2 SRI model and computation using semantic models

While the models presented above allow for the representation of different aspects of a building, none of them cater for the representation of the newly introduce SRI assessments. In this section we describe the SRI model developed within PHOENIX to support the automatic SRI computation, one of the key services that will be offered by the PHOENIX platform.

The Smart Readiness Indicator (SRI)[15] is a tool to promote the use of smart building technologies as foreseen by the European Energy Performance of Buildings Directive (EPBD) in 2018.

The SRI of a building is computed by the following steps:

1. Building assessment, which checks at which level is each applicable smart ready service implemented.
2. Normalisation and computation of scores for each domain-impact combination.
3. Aggregation of these scores to impact criteria.
4. Final aggregation to a single SRI building score.

PHOENIX introduces a Knowledge Graph based approach to apply the SRI methodology. We reuse and apply existing data modelling and data management best practices to ensure high quality data and maximum reusability. Figure 11 shows an overview of the automated PHOENIX SRI computation process. The building assessment (step 1) is represented by the top yellow part. The following steps are represented by the three circles in the middle.

[15] *https://smartreadinessindicator.eu/*

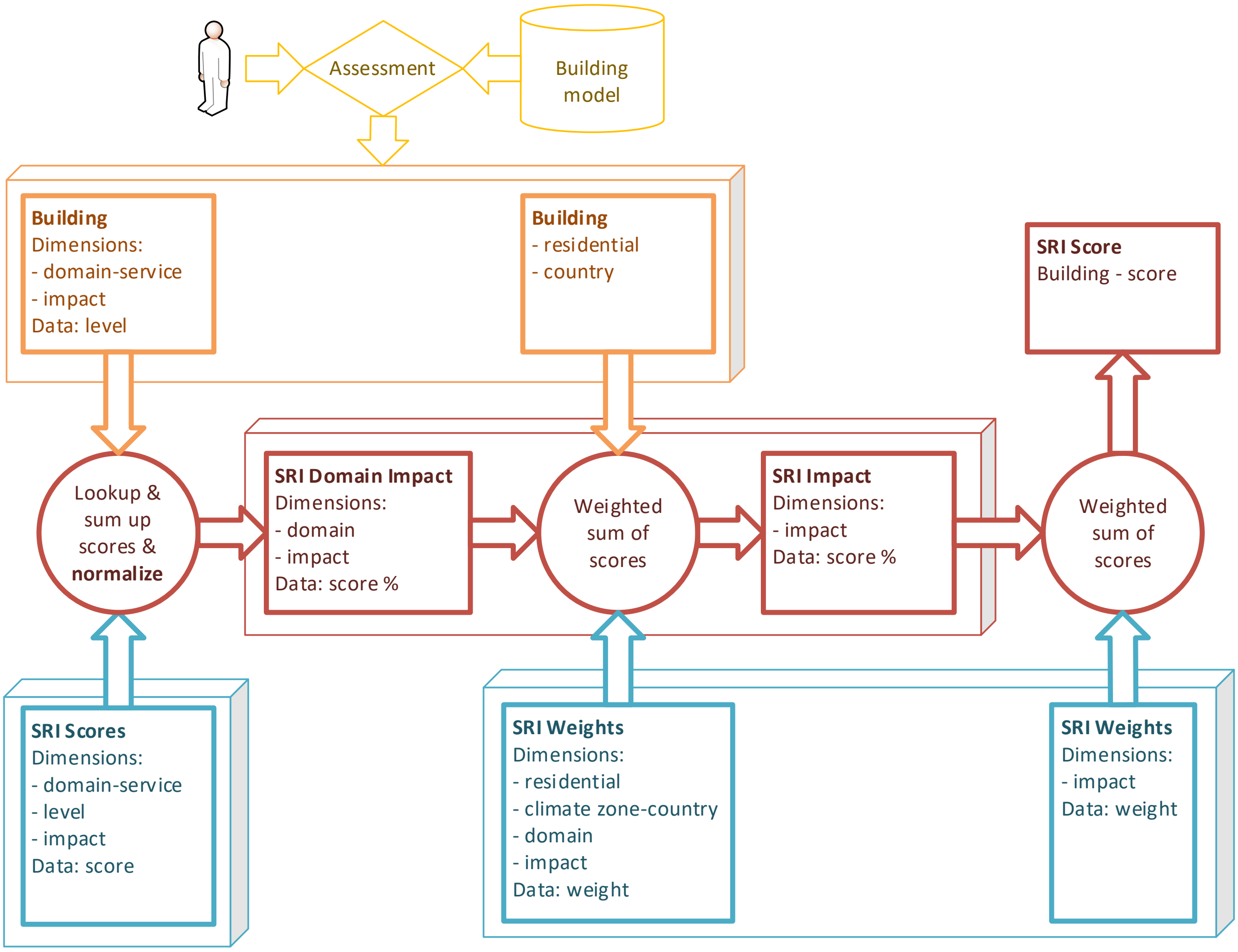


Figure 11 - Overview of the SRI Computation Process.

PHOENIX provides the following building blocks to enable automatic computation of a buildings SRI.

1. Semantic representation of SRI concepts.
2. Semantic multidimensional representation of SRI reference data.
3. Semantic model for representing building assessments.
4. Automated approach to compute SRI impact scores and total SRI score.

These different building blocks for the automatic SRI computation using semantic models are described next.

The **semantic representation** of the SRI computation is formalised as a lightweight Web

Ontology Language (OWL) ontology. This ontology describes the different SRI concepts in a formal manner, mainly smart ready services, domains, functionality levels, impact criteria, key functionalities, scores, and weighing factors.

For the semantic representation of the SRI **reference data**, the mapping of service-levels to scores and the weighing factors, we need a modelling approach which can represent multidimensional data. The rectangles in Figure 11 list the necessary dimensions. For example, the score weight depends on the following 4 dimensions: impact criterion, domain, climate zone and type of the building (residential or non-residential). Furthermore, the model needs to cater for hierarchies: each of the low-granular services are assigned to exactly one coarse-granular domain. Similarly, each country is part of exactly one climate zone and each impact is part of exactly one key functionalities. We model the reference datasets (scores and weights) by using the RDF Data Cube Vocabulary (QB)[16]. QB is a W3C standard created to model multidimensional data in RDF. It is often used for representing statistical data in RDF and is thus well suited to represent multidimensional matrices of numerical data including aggregation hierarchies. We use RDF cube to refer to services, domains, impacts, etc. in these reference datasets. Below we provide examples for the description of the SRI scores, weights, the domain and impact hierarchies and the climate zone.

```
@prefix eg: <example> .
@prefix sri: <https://eu-phoenix.eu/sri#> .
@prefix qb: <http://purl.org/linked-data/cube#> .

# SRI Scores

sri:ohs10  a qb:Observation ; qb:dataSet sris:scoresDS ; sri:function sri:heating-
S1 ; sri:level sri:level0 ; sri:impact sri:energySavingsOnSite ; sri:score 0 .
sri:ohs11  a qb:Observation ; qb:dataSet sris:scoresDS ; sri:function sri:heating-
S1 ; sri:level sri:level1 ; sri:impact sri:energySavingsOnSite ; sri:score 1 .
sri:ohs12  a qb:Observation ; qb:dataSet sris:scoresDS ; sri:function sri:heating-
S1 ; sri:level sri:level2 ; sri:impact sri:energySavingsOnSite ; sri:score 2 .
sri:ohs13  a qb:Observation ; qb:dataSet sris:scoresDS ; sri:function sri:heating-
S1 ; sri:level sri:level3 ; sri:impact sri:energySavingsOnSite ; sri:score 2 .
sri:ohs14  a qb:Observation ; qb:dataSet sris:scoresDS ; sri:function sri:heating-
S1 ; sri:level sri:level4 ; sri:impact sri:energySavingsOnSite ; sri:score 3 .

sri:ohs2a0 a qb:Observation ; qb:dataSet sris:scoresDS ; sri:function sri:heating-
S2a ; sri:level sri:level0 ; sri:impact sri:energySavingsOnSite ; sri:score 0 .
```

[16] *https://www.w3.org/TR/vocab-data-cube/*

```
sri:ohs2a1 a qb:Observation ; qb:dataSet sris:scoresDS ; sri:function sri:heating-S2a ; sri:level sri:level1 ; sri:impact sri:energySavingsOnSite ; sri:score 1 .
sri:ohs2a2 a qb:Observation ; qb:dataSet sris:scoresDS ; sri:function sri:heating-S2a ; sri:level sri:level2 ; sri:impact sri:energySavingsOnSite ; sri:score 2 .

sri:ocs10  a qb:Observation ; qb:dataSet sris:scoresDS ; sri:function sri:cooling-S1 ; sri:level sri:level0 ; sri:impact sri:energySavingsOnSite ; sri:score 0 .
sri:ocs11  a qb:Observation ; qb:dataSet sris:scoresDS ; sri:function sri:cooling-S1 ; sri:level sri:level1 ; sri:impact sri:energySavingsOnSite ; sri:score 1 .
sri:ocs12  a qb:Observation ; qb:dataSet sris:scoresDS ; sri:function sri:cooling-S1 ; sri:level sri:level2 ; sri:impact sri:energySavingsOnSite ; sri:score 1 .
sri:ocs13  a qb:Observation ; qb:dataSet sris:scoresDS ; sri:function sri:cooling-S1 ; sri:level sri:level3 ; sri:impact sri:energySavingsOnSite ; sri:score 2 .
sri:ocs14  a qb:Observation ; qb:dataSet sris:scoresDS ; sri:function sri:cooling-S1 ; sri:level sri:level4 ; sri:impact sri:energySavingsOnSite ; sri:score 3 .

# SRI weights

## Domain - impacts
eg:ow0 a qb:Observation ; qb:dataSet sris:weightsDS ; sri:function sri:heating ; sri:impact sri:energySavingsOnSite ; sri:spatial sri:southEurope ; sri:type sri:residential ; sri:weight 0.32 .
eg:ow1 a qb:Observation ; qb:dataSet sris:weightsDS ; sri:function sri:cooling ; sri:impact sri:energySavingsOnSite ; sri:spatial sri:southEurope ; sri:type sri:residential ; sri:weight 0.07 .

## impacts
eg:owi0 a qb:Observation ; qb:dataSet sris:weightsDS ; sri:function sri:allFunctions ; sri:impact sri:energySavingsAndOperation ; sri:spatial sri:allZones ; sri:type sri:allTypes ; sri:weight 0.167 .

# SRI hierarchies

## Domains and services

sri:heating a sri:Domain ;
   sri:hasPart sri:heating-S1 , sri:heating-S2a .
sri:cooling a sri:Domain ;
   sri:hasPart sri:cooling-S1 .

sri:heating-S1 a sri:Service .
sri:heating-S2a a sri:Service .
sri:cooling-S1 a sri:Service .

## Technical capabilities and impacts

sri:energySavingsAndOperation a sri:Capability ;
   sri:hasPart sri:energySavingsOnSite .

sri:energySavingsOnSite a sri:Impact .
```

```
## Countries and climate zones

sri:southEurope a sri:ClimateZone ;
  sri:hasPart eg:Spain .
```

For representing **building assessments,** we also use the terms of the ontology in a similar manner. Similar to the reference data, QB serves as a basis for the building assessments, in this case with the services and impacts dimensions, and a level as value. Part of a building assessment in RDF using Turtle notation[17] can be represent as follows:

```
@prefix assessment: <http://example.com/assessment#> .
@prefix sri: <https://eu-phoenix.eu/sri#> .
@prefix qb: <http://purl.org/linked-data/cube#> .

# Building assessment

eg:o00 a qb:Observation ; qb:dataSet assessment:a4398 ; sri:function sri:heating-S1  ; sri:level sri:level2 .
eg:o01 a qb:Observation ; qb:dataSet assessment:a4398 ; sri:function sri:heating-S2a ; sri:level sri:level1 .
eg:o02 a qb:Observation ; qb:dataSet assessment:a4398 ; sri:function sri:cooling-S1  ; sri:level sri:level1 .

assessment:a4398 sri:hasBuilding assessment:building4398 .
assessment:building4398 sri:inCountry assessment:Spain ;
  sri:hasType sri:residential .
```

After a building is assessed, this assessment data is inserted into the PHOENIX Building Knowledge Graph where the reference data is already stored. To compute the matrices, domain-impact scores and impact scores, as well as the final building SRI score, only three predefined update queries have to be executed. These three queries correspond to the three different operations (circles in Figure 11). By using a knowledge graph representation, not only can we support the automatic SRI score computation, but we are also given the flexibility to easily adapt to eventual changes in the SRI methodology. Next, we focus on the status of the data analytics services.

5 Data analytics services

All the data analytics services delivered by the PHOENIX Platform are enabled in the Function Layer of the PHOENIX architecture. The Function Layer offers smart cost-effective services with user-friendly interfaces addressed on non-technical end-users (e.g., building owners and

[17] *https://www.w3.org/TR/turtle/*

occupants) and technical stakeholders (e.g., ESCOs). The Function Layer is fed by the Knowledge Layer, where the actual data algorithms for user-centric and grid-centric services are developed, to enable self-learning capabilities and automated decisions to improve energy savings and the overall energy performance of the building. Figure 12 demonstrates the relationship of the two layers.

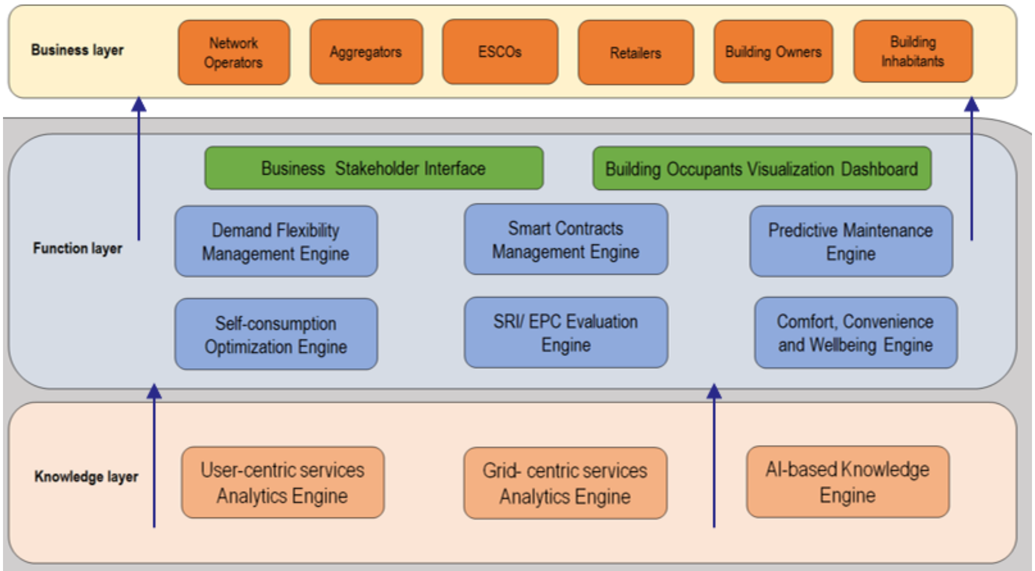

Business layer
Network Operators
Aggregators
ESCOs
Retailers
Building Owners
Building Inhabitants
Function layer
Business Stakeholder Interface
Building Occupants Visualization Dashboard
Demand Flexibility Management Engine
Smart Contracts Management Engine
Predictive Maintenance Engine
Self-consumption Optimization Engine
SRI/ EPC Evaluation Engine
Comfort, Convenience and Wellbeing Engine
Knowledge layer
User-centric services Analytics Engine
Grid- centric services Analytics Engine
AI-based Knowledge Engine

Figure 12 - The Knowledge layer feeds the Function layer.

The three main components of the Knowledge layer are the User-centric services Analytics Engine, the Grid-centric services Analytics Engine and the AI-based Knowledge Engine. The AI-based Knowledge Engine will be enabled by the Knowledge Graph introduced in Section 0. The user-centric engine will consist of three modules, that will be described next. These are currently under development. The following section will then focus on the grid-centric engine, where a small grid is being simulated in order to test the algorithms. It is worth to note that both the user-centric and the grid-centric services engines are not working in silos, but their goal is to provide synergies towards a well-established comfort, convenient, energy efficient and flexible environment that promotes a smart user-building-grid relationship with a goal to promote automatic energy savings.

5.1 Data analytics for user-centric services

The User-centric services Analytics Engine (Figure 13) will actively contribute to the provision of data analytics tools towards the assurance of user-centric services and user comfort-profile preferences. On this basis, the initial scope of the user-centric services is to derive optimal settings on the comfort profiles for each individual building occupant through their initial user profiles (provided at each user registration process step), their dynamic actions on sensor devices in the indoor environment, as well as their corresponding direct feedback to the Building Occupants Visualization Dashboard (if any) and the recognition of viable and customized indoor occupancy conditions. Additionally, energy descriptive and predictive analytics will be provided, in order to deliver useful insights to the building occupant, regarding his indoor environment for better awareness and assurance of optimal indoor conditions. For that we are developing three different modules within the user-centric engine, which are described next.

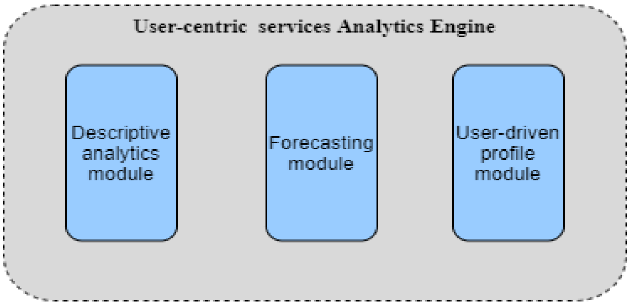

User-centric services Analytics Engine
Descriptive analytics module
Forecasting module
User-driven profile module

Figure 13 - The different modules responsible for the different data algorithms for the user-centric services.

The **user-driven profile module** is responsible for calculating and updating the user profile thermal, visual and air quality preferences. As a first step, the new user registers to the PHOENIX User Dashboard, reads the terms and conditions on their data usage and accepts the terms and conditions (if he rejects, he cannot participate in the program). Once terms are accepted, he is redirected to complete a survey consisting of a set of questions that will help build his default user profile. The list of these questions will be presented and analysed in Deliverable 5.1 – "Services for building's occupants".

From an algorithmic point of view, what is worth mentioning is the initialization of the min/max comfort values on the indoor temperature (thermal profile), on the indoor air quality (IAQ profile) and on the indoor illuminance (visual profile). This initial state, let us call it default state from now on, is stored in the User Registration Database upon the initial registration.

In the first simple and single-parameter approach, and let us take the example of the measurement of indoor temperature T, if the presence sensor does not detect anyone in the room (occupancy = 0), then we rely on these default min/max limits and we do not have any comfort or discomfort events to identify and further analyse. In case of human presence (occupancy = 1), we can either have a comfort event (the user does not manually change the temperature on a given threshold of

0.5 °C) or a discomfort event (the user manually changes the temperature, $\Delta T > 0.5$ °C). Similar methodology applies for the air quality and lighting.

In the multi-parametric approach, again for the example of the thermal profile, the concept is that the user comfort does not only rely on the measured temperature, but it also depends on other factors such as indoor humidity (seasonality effect between summer and winter), CO_2 concentration, room size, people concentration, etc. In this case, a supervised machine learning technique can be applied, initially on multiple training comfort and discomfort event data sets (probably per dwelling) and then test the model on new event data. The labelled data events are fed into the Comfort, Convenience and Wellbeing Engine that will eventually trigger either a notification event or an automated control event for the user.

The **descriptive analytics module** is responsible to deliver raw real time, aggregated and historic data results in order for the user to be constantly aware of his environment, compare with past measurements, similar peers, etc. More specifically the module is responsible for:

- Aggregated and historical analytics on Temperature, CO2, humidity, illuminance, and external weather conditions.
- Total and average building energy consumption, consumption of similar peers and comparative past performance on energy consumption.
- Aggregated and historical analytics on energy generation and storage (if applicable in the prosumer case).
- Total and average building energy waste considering the cooling and heating degree days.
- Relation and patterns of energy consumption with environmental conditions, user actions, demographics, and daily routines.

The **forecasting analytics module** is responsible to deliver a set of predictive algorithms for multiple cases, including:

- Predictions on how the energy consumption is related to other environmental variables.
- Predictions on the impact of the notifications and recommendations interventions on energy consumption.
- Predictions on the impact of user actions/interactions with the platform on energy consumption.

As we mentioned before, these three models will be used in combination to offer user-centric services geared to comfort optimization, considering multiple data inputs, both users' direct actions (e.g., profile and active temperate settings) and passive data coming from sensors installed in the building. The user experience will also be enhanced by delivering insightful forecasts to the users, for better awareness and assurance of optimal indoor conditions. In the next section we describe the status of the analytical engine geared towards the grid.

5.2 Data analytics for grid integration services

Interactions with the grid are a key element of PHOENIX. These interactions will be done thanks to decision making deeply rooted on the Smartness Hub of the project. There are two main families of algorithms that have been identified in this aspect. The algorithms for building's optimal operation based on information coming from the grid and other external data sources (such as tariff markets or weather), and the algorithms to design and deploy demand response events. This is summarised in the following table.

Table 1 - Two different families of algorithms that will take part of Grid integration services within PHOENIX.

Algorithms for optimal operation by "listening" to the grid	Algorithms for two-way communication with the grid
- Energy saving recommendation - Rescheduling of devices operation - Optimal use of Electric Vehicle charging - Personalised feedback - Peak load shifting - Grid-related energy literacy	- Demand response events - Pro/consumers - Optimal transaction operation - Direct load control - Power Infrastructure recommendation

The PHOENIX platform will have a series of components that will allow to perform the intelligence needed to develop advanced services on this respect.

Grids are rather complex systems, and, in most cases, it is difficult to fully monitor what is happening at them, in terms of consumption and production. For the PoC, we have created a synthetic micro-grid for better deployment and testing of services. This micro-grid has been equipped with simulated humans that, following probability distributions obtained from real data, occupy their buildings on a realistic manner. Also, several appliances have been modelled, and they are used also following realistic probability distributions. Among these appliances, one can find wet appliances (washing machine and dishwasher), entertainment appliances (TV or computer) and constant use appliances such as refrigerators.

The synthetic framework created had buildings located on realistic positions, and with realistic thermal behaviour (thermal inertia) to make possible the validation of algorithms investigating the optimal flows of energy between buildings that may act as consumers or producers depending on the given instant, the grid situation, or the status of the electrical storage (Figure 14).

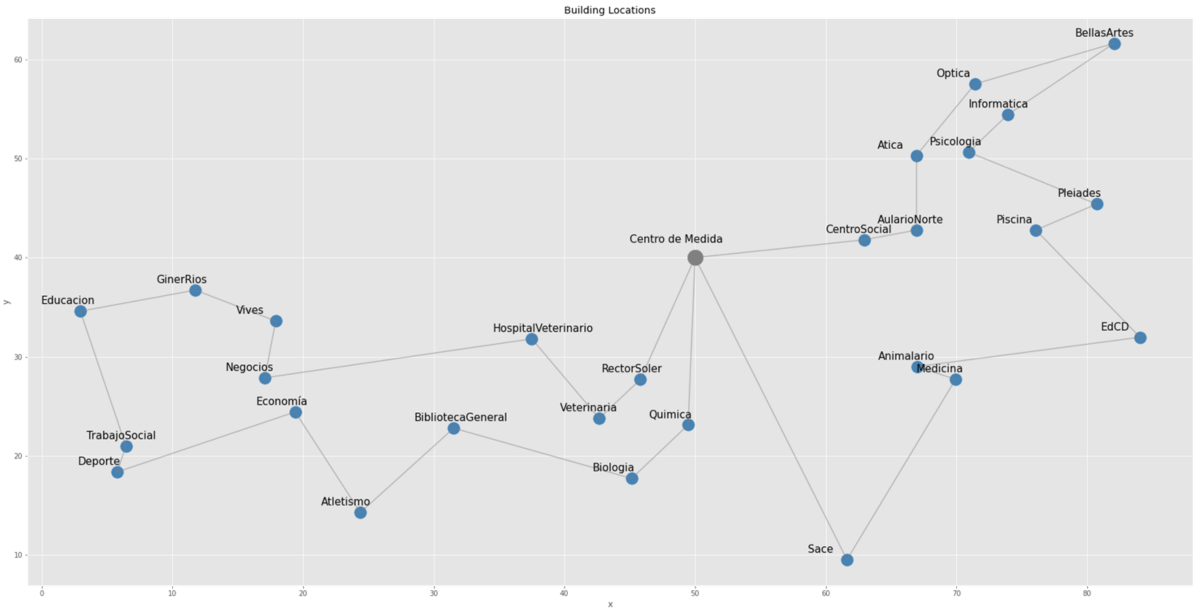


Figure 14 - Map view of the neighborhood with a series of nodes. To have a realistic electrical network, the medium voltage grid of the University of Murcia has been used.

In addition to the sensor data, the meters, the weather information, and the framework for the synthetic simulation of the grid, the services for grid integration will also have the connectivity with the API of *Red Eléctrica Española* (REE) (Figure 15). REE is the Transmission System Operator (TSO), of Spain, although it operates on other countries as well.

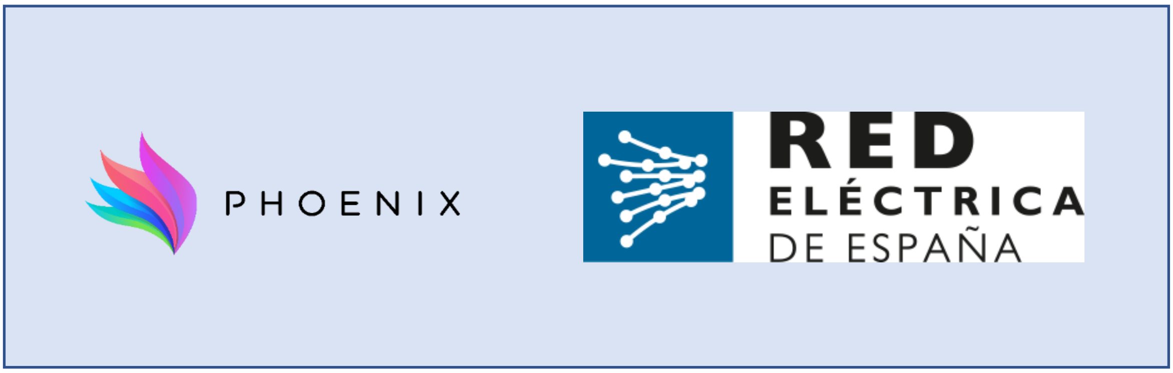


Figure 15 - Integration de PHOENIX with Red Electrica Española.

The API of REE offers the possibility of consulting in real time the mix of production for that given moment opening the door to use AI to optimise not only the scheduling for reducing peaks in kilowatts, but also on reducing peaks in CO_2 emissions. An example of the output that one can get from the API of REE can be found on Figure 16.

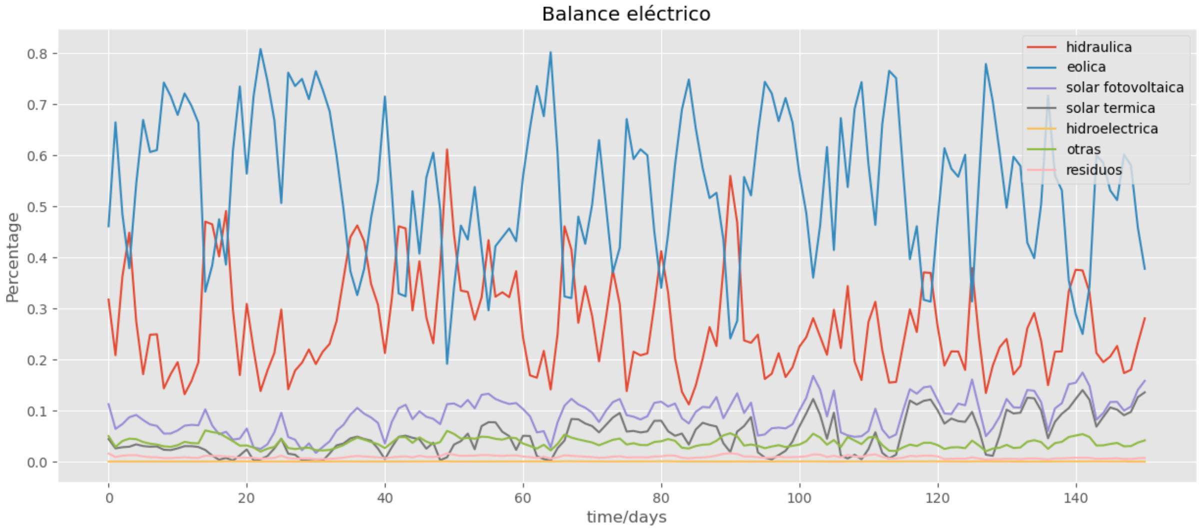
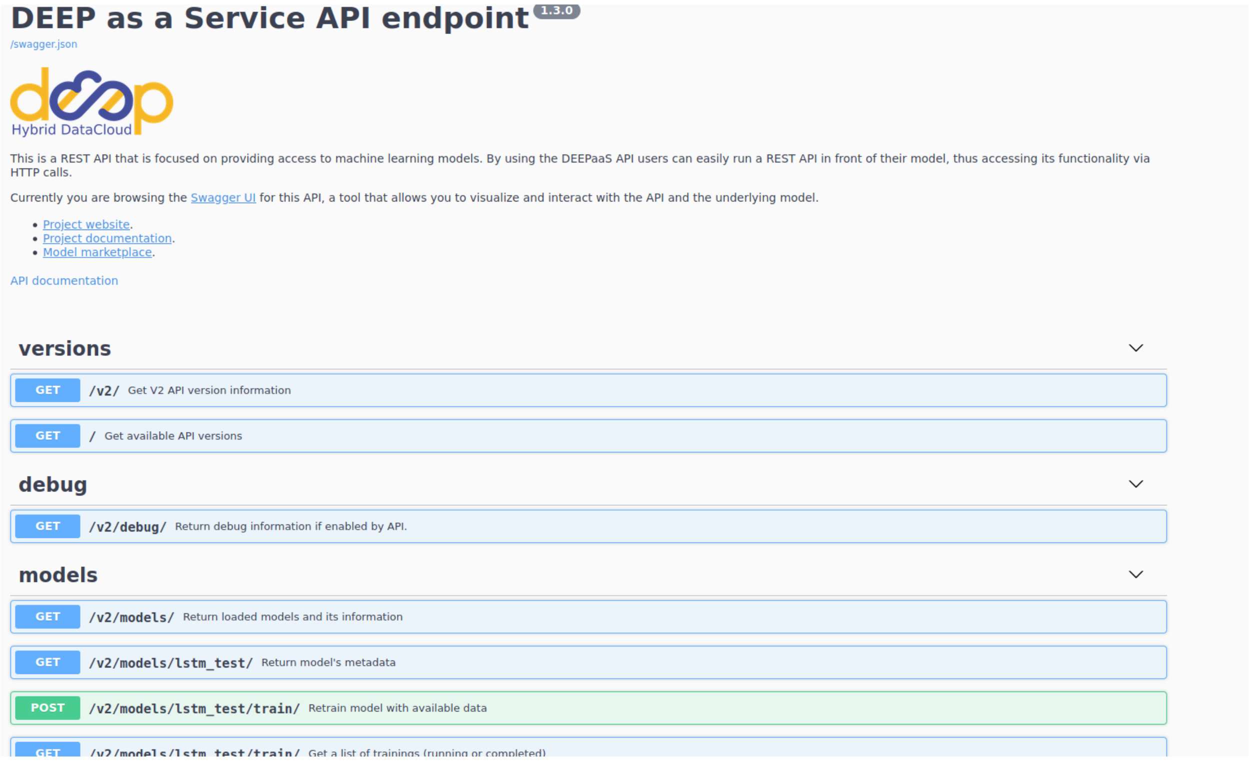


Figure 16 - Representation of renewable production for 150 days using the REE API. The text (in Spanish on the figure) reads: Electric Balance, Hydraulic, Wind power, Solar PV, Solar thermal, Hydroelectric, others, garbage.

We are currently developing and testing different AI algorithms for energy availability forecasting based on the data sources described above. For developing the algorithms that will need High-Performance Computing capabilities, we are using the Deep Hybrid DataCloud[18] platform for

[18] *https://deep-hybrid-datacloud.eu/*

univariate series (ARIMA) and multivariate scenarios (LSTM). This allows us to make the models available through the DEEP as a Service (DEEPaaS) REST API [19] in order to input the data to them and trigger the training and testing phases by means of HTTP requests as it can be seen in Figure 17.



DEEP as a Service API endpoint 1.3.0

/swagger.json

Hybrid DataCloud

This is a REST API that is focused on providing access to machine learning models. By using the DEEPaaS API users can easily run a REST API in front of their model, thus accessing its functionality via HTTP calls.

Currently you are browsing the Swagger UI for this API, a tool that allows you to visualize and interact with the API and the underlying model.

- Project website.
- Project documentation.
- Model marketplace.

API documentation

versions

GET /v2/ Get V2 API version information

GET / Get available API versions

debug

GET /v2/debug/ Return debug information if enabled by API.

models

GET /v2/models/ Return loaded models and its information

GET /v2/models/lstm_test/ Return model's metadata

POST /v2/models/lstm_test/train/ Retrain model with available data

GET /v2/models/lstm_test/train/ Get a list of trainings (running or completed)

Figure 17 - DEEP as a Service API to train and test the developed algorithms.

So far, we have described the data available for the development of the PoC, the models in use and under consideration, the different data storages, as well as the analytical engines being developed to support both user and grid-oriented services. Orthogonal to the work presented until now, PHOENIX is developing and implementing a number of features to ensure the security and privacy of the data. The security & privacy framework is detailed in the next section. In particular, we describe the risk analysis performed at the early stage of the project, which lead to a series of features to be implemented. The section also includes a review of the identified features and how they are instantiated within the PHOENIX Smartness Hub.

[19] *García, Á. L. (2019). DEEPaaS API: A REST API for machine learning and deep learning models. Journal of Open Source Software, 4(42), 1517.*

6 Security and privacy features

This section presents the activities so far to provide security and privacy mechanisms to the PHOENIX solution. To achieve that, a security & privacy (S&P) framework has been designed as a proxy between IoT devices in the buildings and external applications, as well as the building Smartness Hub at the cloud. This framework aims to protect the data from IoT sensors and smart equipment in the building against unauthorised access within the building. At first, we have performed a security and privacy analysis of risks and requirements in distributed IoT scenarios, in particular smart building section. According to the risks and requirements, we have identified security mechanism as well as contingency measures to mitigate the risks. In the following subsections, we describe in detail the requirements analysis and the mechanisms identified and integrated to protect the PHOENIX ecosystem.

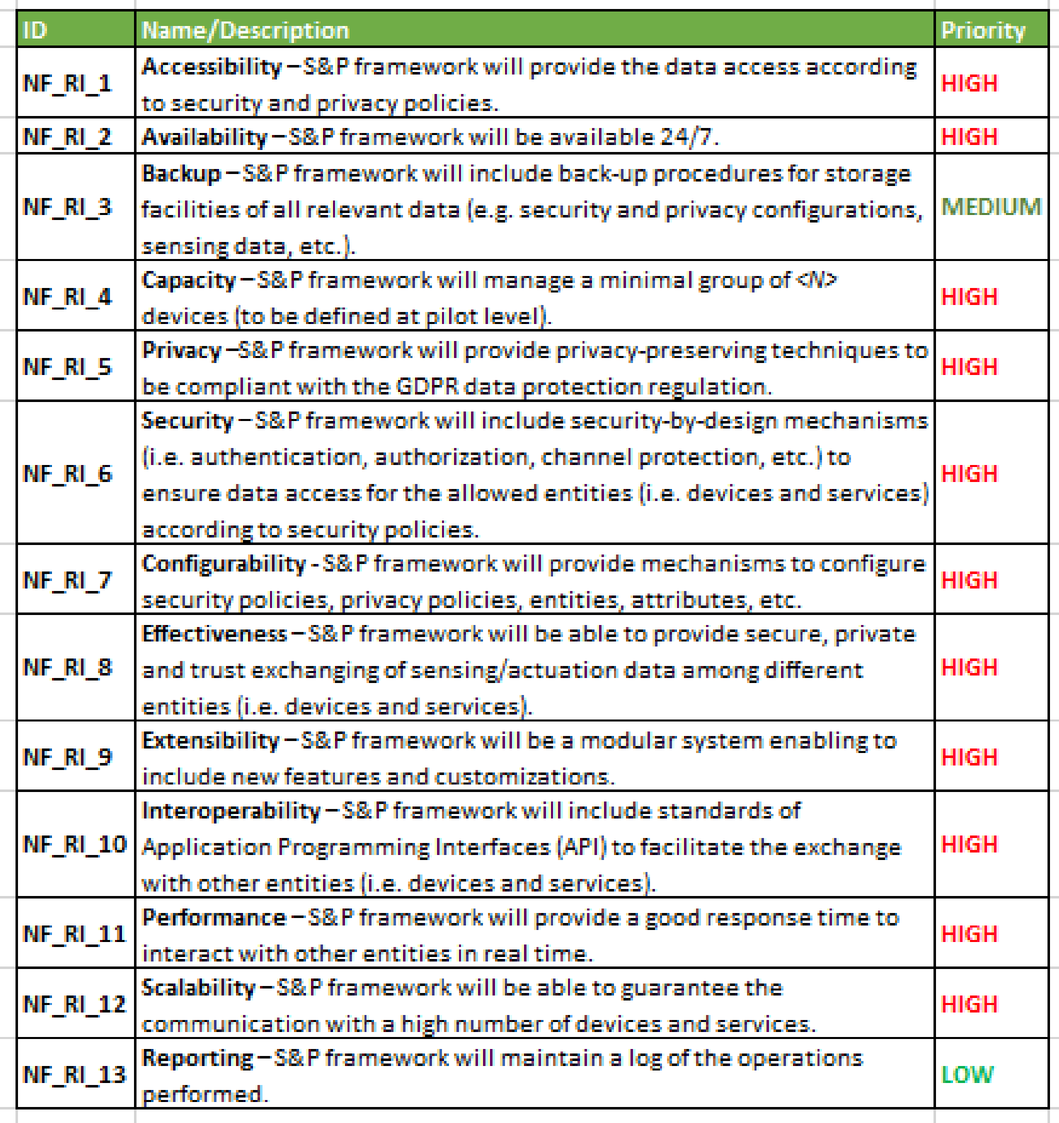
6.1 Risk analysis associated to privacy/security features

At the early stages of the project, we have performed an analysis of the technical risks related to privacy, security, trust features in IoT and Smart Building ecosystems. The outcomes of this analysis are described below. More specifically, we list all identified risks for two different categories, namely non-functional and functional risks.

6.1.1 Non-functional risks

Table 2 lists the identified non-functional risks associated to Smart Building scenarios and security/privacy features that the S&P framework should deal with. The risks were giving an identifier to better refer to them in the future. Moreover, each risk was assigned to one of three different levels of priority (LOW, MEDIUM , and HIGH).

Table 2 - Non-functional risks in PHOENIX.

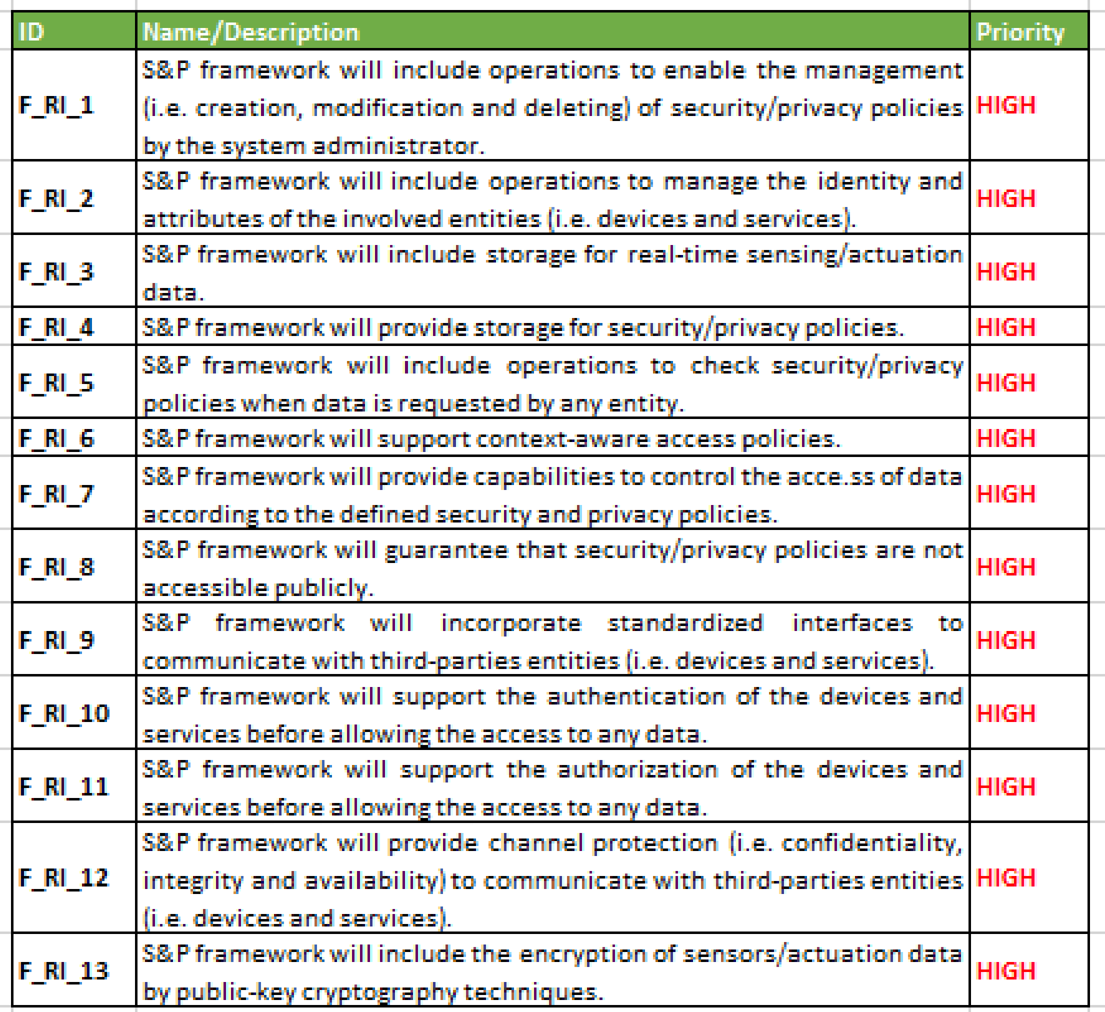


ID	Name/Description	Priority
NF_RI_1	**Accessibility** – S&P framework will provide the data access according to security and privacy policies.	HIGH
NF_RI_2	**Availability** – S&P framework will be available 24/7.	HIGH
NF_RI_3	**Backup** – S&P framework will include back-up procedures for storage facilities of all relevant data (e.g. security and privacy configurations, sensing data, etc.).	MEDIUM
NF_RI_4	**Capacity** – S&P framework will manage a minimal group of <N> devices (to be defined at pilot level).	HIGH
NF_RI_5	**Privacy** –S&P framework will provide privacy-preserving techniques to be compliant with the GDPR data protection regulation.	HIGH
NF_RI_6	**Security** – S&P framework will include security-by-design mechanisms (i.e. authentication, authorization, channel protection, etc.) to ensure data access for the allowed entities (i.e. devices and services) according to security policies.	HIGH
NF_RI_7	**Configurability** - S&P framework will provide mechanisms to configure security policies, privacy policies, entities, attributes, etc.	HIGH
NF_RI_8	**Effectiveness** – S&P framework will be able to provide secure, private and trust exchanging of sensing/actuation data among different entities (i.e. devices and services).	HIGH
NF_RI_9	**Extensibility** – S&P framework will be a modular system enabling to include new features and customizations.	HIGH
NF_RI_10	**Interoperability** – S&P framework will include standards of Application Programming Interfaces (API) to facilitate the exchange with other entities (i.e. devices and services).	HIGH
NF_RI_11	**Performance** – S&P framework will provide a good response time to interact with other entities in real time.	HIGH
NF_RI_12	**Scalability** – S&P framework will be able to guarantee the communication with a high number of devices and services.	HIGH
NF_RI_13	**Reporting** – S&P framework will maintain a log of the operations performed.	LOW

6.1.2 Functional risks

Table 3 lists the functional risks in the context of security/privacy features that should be addressed by the PHOENIX solution for advancing in protected Smart Building scenarios. As with the non-functional risk, each functional risk was given an identified and classified according to different levels of priority (LOW, MEDIUM , and HIGH).

Table 3 - Functional requirements of the S&P framework.



ID	Name/Description	Priority
F_RI_1	S&P framework will include operations to enable the management (i.e. creation, modification and deleting) of security/privacy policies by the system administrator.	HIGH
F_RI_2	S&P framework will include operations to manage the identity and attributes of the involved entities (i.e. devices and services).	HIGH
F_RI_3	S&P framework will include storage for real-time sensing/actuation data.	HIGH
F_RI_4	S&P framework will provide storage for security/privacy policies.	HIGH
F_RI_5	S&P framework will include operations to check security/privacy policies when data is requested by any entity.	HIGH
F_RI_6	S&P framework will support context-aware access policies.	HIGH
F_RI_7	S&P framework will provide capabilities to control the acce.ss of data according to the defined security and privacy policies.	HIGH
F_RI_8	S&P framework will guarantee that security/privacy policies are not accessible publicly.	HIGH
F_RI_9	S&P framework will incorporate standardized interfaces to communicate with third-parties entities (i.e. devices and services).	HIGH
F_RI_10	S&P framework will support the authentication of the devices and services before allowing the access to any data.	HIGH
F_RI_11	S&P framework will support the authorization of the devices and services before allowing the access to any data.	HIGH
F_RI_12	S&P framework will provide channel protection (i.e. confidentiality, integrity and availability) to communicate with third-parties entities (i.e. devices and services).	HIGH
F_RI_13	S&P framework will include the encryption of sensors/actuation data by public-key cryptography techniques.	HIGH

Based on the risk analysis, we have identified different components and mechanisms that need to be integrated and extended in order to guarantee the security & privacy in the internet communications among IoT gateways located in the building pilots and ICT systems of the PHOENIX Smartness Hub at the cloud. These are presented in the following section.

6.2 Mechanisms for Security & Privacy Framework

This section presents the by-design privacy & security features addressed by the S&P framework. More specifically, the S&P framework enables the following privacy/security features: secure authentication/bootstrapping; anonymized identity management, and authorization & access control. For each of these features, this section describes the requirements and challenges with the implementation of the feature, the existing body of work and their shortcomings, and how

PHOENIX advance state-of-the-art in the IoT security and privacy domain.

6.2.1 Secure Authentication & Bootstrapping

For the bootstrapping process of the initial connection of IoT devices and ICT systems, the secure authentication is required to ensure the identity of any entity (i.e., IoT device or ICT system) is valid. This process allows verifying the identity of any entity that wants to connect to the PHOENIX Smartness Hub. The bootstrapping process enables the authentication of any entity regarding the required credentials. Different credentials can be used, for instance, shared keys, digital certificate, and login/password. The main result of this bootstrapping process is a token of authentication generated by the S&P framework. The entity can then use this token in future interactions (i.e., publishing or retrieving data) with the PHOENIX Smartness Hub, where the token is presented to verify if this concrete entity was authenticated successfully by the S&P framework.

The S&P framework enables a sophisticated method to perform the secure authentication of any entity, as well as allowing minimal disclosure of privacy data about the entity and its identity in next interactions. This privacy-preserving process is achieved thanks to the generation of anonymous identification data generated by the Identity Management (IdM) component. This element creates fresh checks the validity of the anonymous attempt and generates an authentication token. The S&P framework permits a sophisticated method to perform the secure authentication of any entity as well as allows minimal disclosure of privacy data about the entity and its identity in next interactions. This privacy-preserving process is achieved thanks to the generation of anonymous identification data generated by the Identity Management (IdM) component. This element creates fresh checks on the validity of the anonymous attempt and generates an authentication token. The IdM is described in the following subsection.

6.2.2 Identity Management (IdM)

In order to retain privacy, modern authentication solutions should disclose only the minimal amount of data in IoT environments. This feature was not conceived in traditional identity management solutions. Additionally, solutions based on certificates, e.g., X.509, demand a centralised solution for identity information storage, hosted by the service provider. As a consequence, all the user identification and authentication data are held by the service provider—withdrawing the user control over what private data gets disclosed. Thus, the design of novel IdM solutions must take into consideration the claim of the user to control how the minimum data is disclosed in order to maintain anonymity.

The goal of the IdM within the S&P framework is to employ pioneering technologies for keeping the maximum level of entity anonymity, be it smart devices and ICT services. Thus, the IdM manages private entity data, e.g., identities, credentials, pseudonyms, by providing system administrators with ways to insert or edit entity private details. Due to the massive and heterogeneous nature of IoT scenarios, the innovations of IdM must enable its deployment at distributed and scalable systems. Following these principles, the IdM mechanisms must achieve deployment at constrained hosts, e.g., IoT gateways with small form factor, and be able to collaborate with the rest of the deployment in order to support random bursts of device activity.

6.2.3 Access Control / Authorization

Due to the heterogeneous and massive nature of IoT deployments, authorization mechanisms must leverage on distributed and lightweight technologies. IoT device capabilities range includes severely constrained microcontrollers. Thus, the methodologies employed in non-constrained scenarios are not apt to be used in these scenarios. These established authorization decision standards include Security Assertion Markup Language (SAML) [20], and eXtensible Access Control Markup Language (XACML) [21]. However, due to the constrained nature of IoT, alternative solutions are required. Likewise, the conventional access control models—i.e., Role-Based Access Control (RBAC) [22], and Attributed Based Access Control (ABAC)[23]—do not provide features that enable resource constrained or distributable scenarios, where the computational tasks are spread among different lightweight hosts at different geographical locations. Hence, their fitness for massive IoT scenarios has not been demonstrated. Furthermore, their computational requirements also make them inappropriate for IoT entities.

In order to address all the aforementioned challenges, the S&P framework integrates novel technologies that enable distributed and scalable solutions by design. In IoT scenarios, these technologies leverage access control via authorization policies and distributed access control

[20] *SAML (Security Assertion Markup Language): http://docs.oasis-open.org/security/saml/v2.0*

[21] *OASIS Standard. eXtensible Access Control Markup Language (XACML) Version 3.0 Plus Errata 01. July 2017: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-en.pdf*

[22] *Ferraiolo, D., Cugini, J., & Kuhn, D. R. (1995, December). Role-based access control (RBAC): Features and motivations. In Proceedings of 11th annual computer security application conference (pp. 241-48).*

[23] *Yuan, E., & Tong, J. (2005, July). Attributed based access control (ABAC) for web services. In Web Services, 2005. ICWS 2005. Proceedings. 2005 IEEE International Conference on. IEEE.*

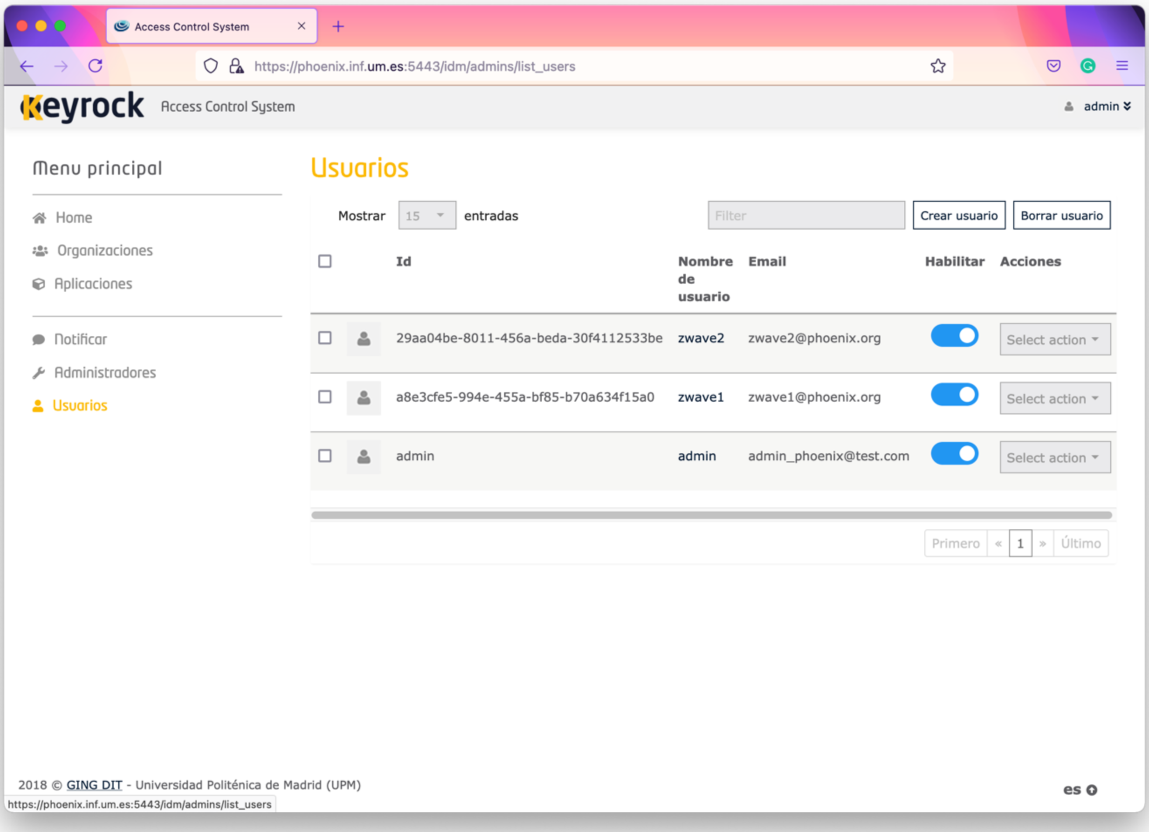
tokens—thus enabling their processing by constrained end-devices and ICT entities. In [24], the authors present two distributed capability-based authorization mechanisms leveraging on access control policies. These security mechanisms have been adopted by the S&P system due to their qualities, designed to take part in the authorization of heterogeneous entities in a distributed manner. Some of these design choices include the use of JSON format and the relatively lightweight ECC-based sign algorithms.

6.3 PoC security components integration: Z-wave sensors gateway

For the case of the gateway in charge of gathering the sensory information from Z-wave sensors, a first integration of the security components has already been performed. The Z-wave gateway's mission is to receive information from the Z-wave network of sensors, and process and format it so that it complies with the PHOENIX data-model, as well as managing the creation and update of the sensor-related entities in the NGSI-LD broker of the architecture. In order to access that broker in a secure manner, the Z-wave gateway needs to follow the complete process of login-in with the Identity Management component (Keyrock) in order to retrieve an Id Token, asking for a Capability Token to the Capability Manager (using the previously obtained Id Token as part of the request) and finally interacting with the broker through the PEP Proxy, attaching the obtained Capability Token, which will provide enough information to the PEP Proxy to decide whether that interaction lies within the boundaries of the policies defined in the system.

Prior to this interaction, a set-up process must be followed, by which the Z-wave gateway's Keyrock users are going to be defined and configured via Keyrock's web GUI, as showcased in Figure 18, where two users can be seen, representing two different Z-wave gateways: zwave1 and zwave2.

[24] *J. L. Hernández-Ramos, A. J. Jara, L. Marín, and A. F. Skarmeta, "Dcapbac: Embedding authorization logic into smart things through ecc optimizations," International Journal of Computer Mathematics, no. just-accepted, pp. 1–22, 2014.*



Access Control System
https://phoenix.inf.um.es:5443/idm/admins/list_users
Keyrock Access Control System
admin
Menu principal
Home
Organizaciones
Aplicaciones
Notificar
Administradores
Usuarios

Usuarios

Mostrar 15 entradas

Filter | Crear usuario | Borrar usuario

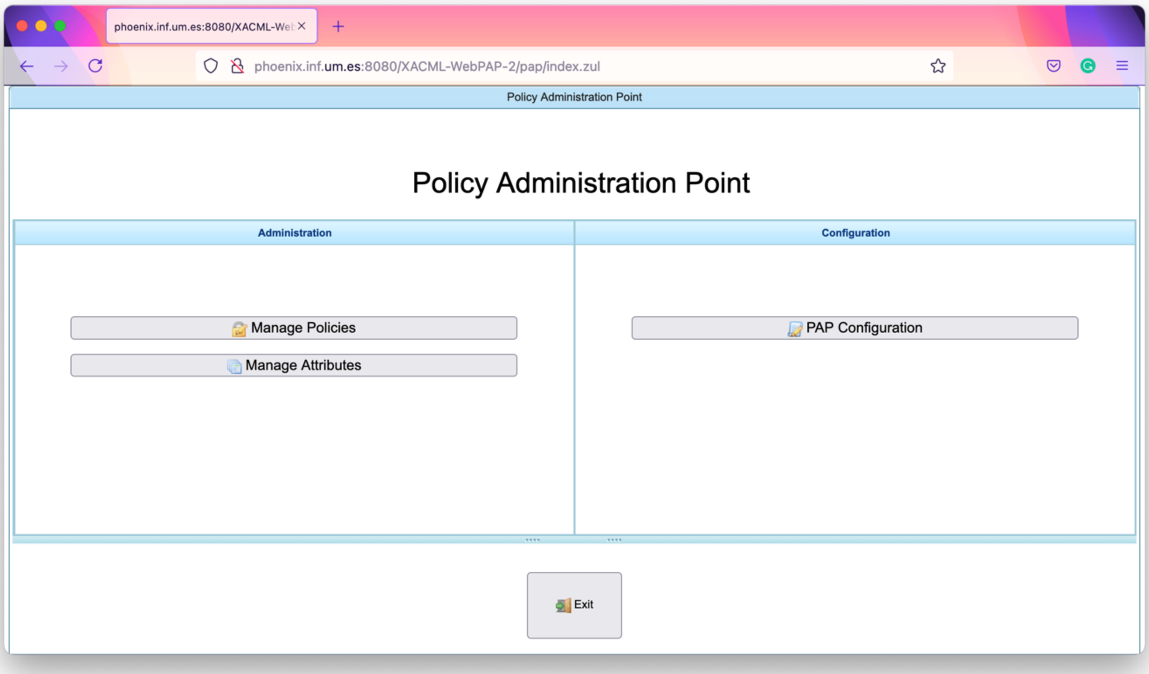
□	Id	Nombre de usuario	Email	Habilitar	Acciones
□	29aa04be-8011-456a-beda-30f4112533be	zwave2	zwave2@phoenix.org	●	Select action
□	a8e3cfe5-994e-455a-bf85-b70a634f15a0	zwave1	zwave1@phoenix.org	●	Select action
□	admin	admin	admin_phoenix@test.com	●	Select action

Primero « 1 » Último
2018 © GING DIT - Universidad Politécnica de Madrid (UPM)
es
https://phoenix.inf.um.es:5443/idm/admins/list_users

Figure 18: IdM - Users

Later we have to define the policies which will define who, how and what can be accessed via the PEP Proxy. In order to do that we use the Policy Administration Point (PAP) GUI, represented in Figure 19, which offers us two forms: one for managing Attributes and other for managing

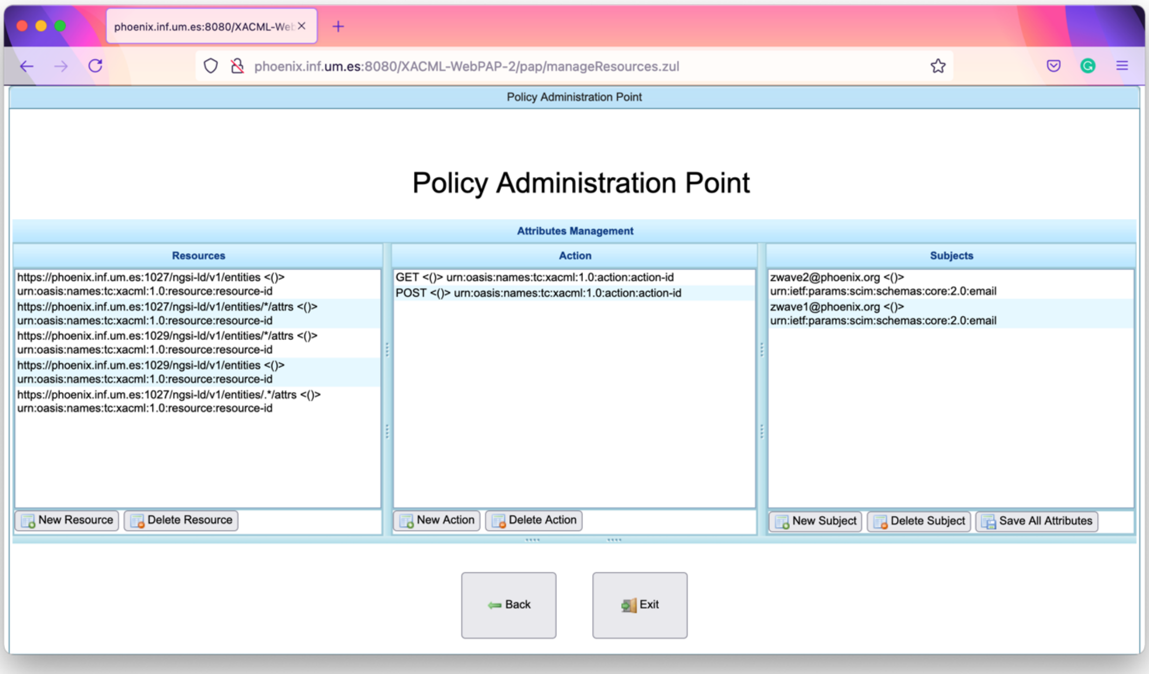
Policies.



phoenix.inf.um.es:8080/XACML-We
phoenix.inf.um.es:8080/XACML-WebPAP-2/pap/index.zul
Policy Administration Point
Policy Administration Point
Administration
Configuration
Manage Policies
Manage Attributes
PAP Configuration
Exit

Figure 19: Policy Administration Point (PAP) GUI

The Attribute management form, shown in Figure 20, will be the entry point of our policy definition process. It is in this point where we will define in the system what can be managed in the form of Resources, Actions and Subjects. Resources represent the different entities contained in the broker that will be accessed through the PEP Proxy and are defined as URLs. In this case we can see that the different URLs for editing and creating entities have been already introduced. Actions represent the “verbs” that can be applied; in this case we are most interested in the POST HTTP verb, but also GET has been defined for future use. Finally, Subjects represent users or roles that will be later be given access to Resources and Actions through the PEP Proxy. As can be seen, both zwave1 and zwave2 have been introduced in the Subjects panel.


phoenix.inf.um.es:8080/XACML-Wel
phoenix.inf.um.es:8080/XACML-WebPAP-2/pap/manageResources.zul
Policy Administration Point
New Resource
Delete Resource
New Action
Delete Action
New Subject
Delete Subject
Save All Attributes
Back
Exit

Policy Administration Point

Attributes Management

Resources

https://phoenix.inf.um.es:1027/ngsi-ld/v1/entities <()> urn:oasis:names:tc:xacml:1.0:resource:resource-id
https://phoenix.inf.um.es:1027/ngsi-ld/v1/entities/*/attrs <()> urn:oasis:names:tc:xacml:1.0:resource:resource-id
https://phoenix.inf.um.es:1029/ngsi-ld/v1/entities/*/attrs <()> urn:oasis:names:tc:xacml:1.0:resource:resource-id
https://phoenix.inf.um.es:1029/ngsi-ld/v1/entities <()> urn:oasis:names:tc:xacml:1.0:resource:resource-id
https://phoenix.inf.um.es:1027/ngsi-ld/v1/entities/.*/attrs <()> urn:oasis:names:tc:xacml:1.0:resource:resource-id

Action

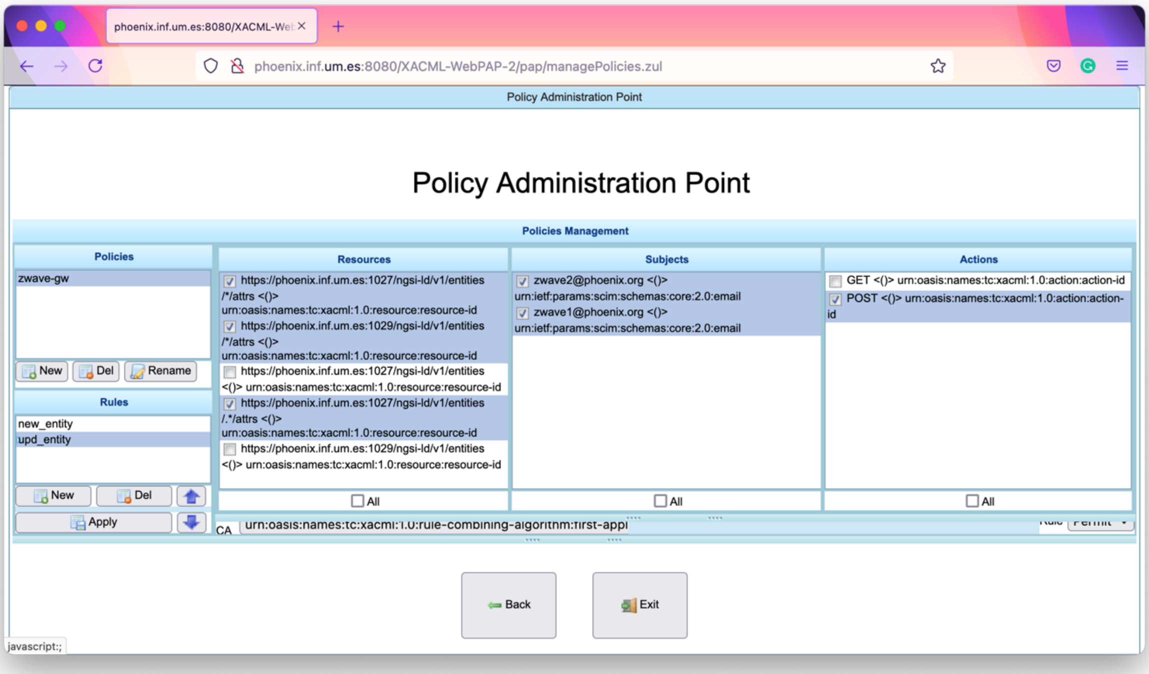
GET <()> urn:oasis:names:tc:xacml:1.0:action:action-id
POST <()> urn:oasis:names:tc:xacml:1.0:action:action-id

Subjects

zwave2@phoenix.org <()> urn:ietf:params:scim:schemas:core:2.0:email
zwave1@phoenix.org <()> urn:ietf:params:scim:schemas:core:2.0:email

Figure 20: PAP - Attributes Management

Once all the elements to build policies from have been introduced in the system, we navigate to the Policies Management panel, and build the different policies that will define the different permissions that will govern the interactions between the Z-wave gateways and the Broker via the PEP Proxy. A policy consists of a set of one or several resources, Actions and Subjects that will define who will access what and how will it be accessed. Figure 21 shows the policy "upd_entity" in which the URL for entity update, the verb "POST" and both "zwave1" and "zwave2" users have been selected, giving access to both Z-wave gateways to posting modifications to entities. Similarly Figure 22 shows the elements involved for giving access to both Z-wave gateways to the entity update mechanism.


phoenix.inf.um.es:8080/XACML-We
phoenix.inf.um.es:8080/XACML-WebPAP-2/pap/managePolicies.zul
Policy Administration Point
New
Del
Rename
New
Del
Apply
Back
Exit
javascript:;

Policy Administration Point

Policies Management

Policies: zwave-gw

Rules: new_entity, upd_entity

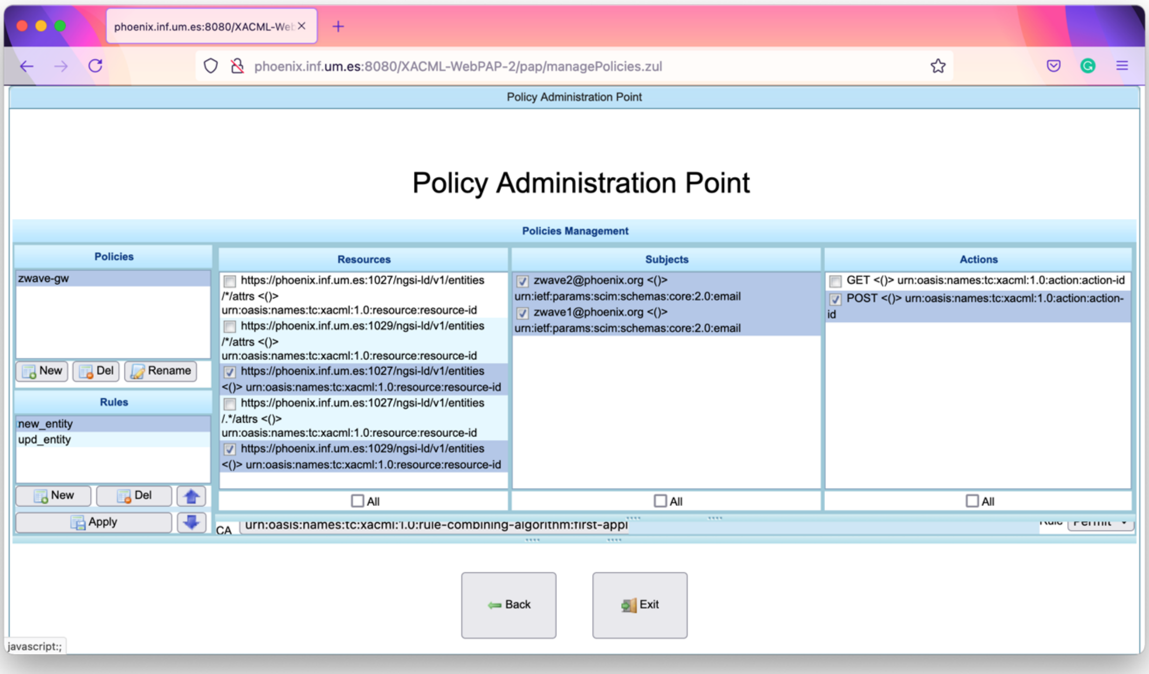
Resources:
- ☑ https://phoenix.inf.um.es:1027/ngsi-ld/v1/entities /*/attrs <()> urn:oasis:names:tc:xacml:1.0:resource:resource-id
- ☑ https://phoenix.inf.um.es:1029/ngsi-ld/v1/entities /*/attrs <()> urn:oasis:names:tc:xacml:1.0:resource:resource-id
- □ https://phoenix.inf.um.es:1027/ngsi-ld/v1/entities <()> urn:oasis:names:tc:xacml:1.0:resource:resource-id
- ☑ https://phoenix.inf.um.es:1027/ngsi-ld/v1/entities /.*/attrs <()> urn:oasis:names:tc:xacml:1.0:resource:resource-id
- □ https://phoenix.inf.um.es:1029/ngsi-ld/v1/entities <()> urn:oasis:names:tc:xacml:1.0:resource:resource-id
- □ All

Subjects:
- ☑ zwave2@phoenix.org <()> urn:ietf:params:scim:schemas:core:2.0:email
- ☑ zwave1@phoenix.org <()> urn:ietf:params:scim:schemas:core:2.0:email
- □ All

Actions:
- □ GET <()> urn:oasis:names:tc:xacml:1.0:action:action-id
- ☑ POST <()> urn:oasis:names:tc:xacml:1.0:action:action-id
- □ All

CA urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-appl
Rule Permit

Figure 21: PAP - Update Entity Policy


phoenix.inf.um.es:8080/XACML-We
phoenix.inf.um.es:8080/XACML-WebPAP-2/pap/managePolicies.zul
Policy Administration Point
New
Del
Rename
New
Del
Apply
Back
Exit
javascript:;

Policy Administration Point

Policies Management

Policies: zwave-gw

Rules: new_entity, upd_entity

Resources:
- □ https://phoenix.inf.um.es:1027/ngsi-ld/v1/entities /*/attrs <()> urn:oasis:names:tc:xacml:1.0:resource:resource-id
- □ https://phoenix.inf.um.es:1029/ngsi-ld/v1/entities /*/attrs <()> urn:oasis:names:tc:xacml:1.0:resource:resource-id
- ☑ https://phoenix.inf.um.es:1027/ngsi-ld/v1/entities <()> urn:oasis:names:tc:xacml:1.0:resource:resource-id
- □ https://phoenix.inf.um.es:1027/ngsi-ld/v1/entities /.*/attrs <()> urn:oasis:names:tc:xacml:1.0:resource:resource-id
- ☑ https://phoenix.inf.um.es:1029/ngsi-ld/v1/entities <()> urn:oasis:names:tc:xacml:1.0:resource:resource-id
- □ All

Subjects:
- ☑ zwave2@phoenix.org <()> urn:ietf:params:scim:schemas:core:2.0:email
- ☑ zwave1@phoenix.org <()> urn:ietf:params:scim:schemas:core:2.0:email
- □ All

Actions:
- □ GET <()> urn:oasis:names:tc:xacml:1.0:action:action-id
- ☑ POST <()> urn:oasis:names:tc:xacml:1.0:action:action-id
- □ All

CA urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-appl
Rule Permit

Figure 22: PAP - New Entity Policy

Lastly, Figure 23 shows the Node Red flow that enables the use of the security components in both Z-wave gateways. As we can see it is a really simple flow in which we can see two processes that fire periodically (every 30 minutes), that refresh the Identity token and the capability tokens for both updating and creating entities. Those tokens are later seamlessly utilized in the "Orion LD" flow in which the communication with the Broker is almost transparently performed via the PEP Proxy by just using the corresponding capability token in the authentication header of the request.

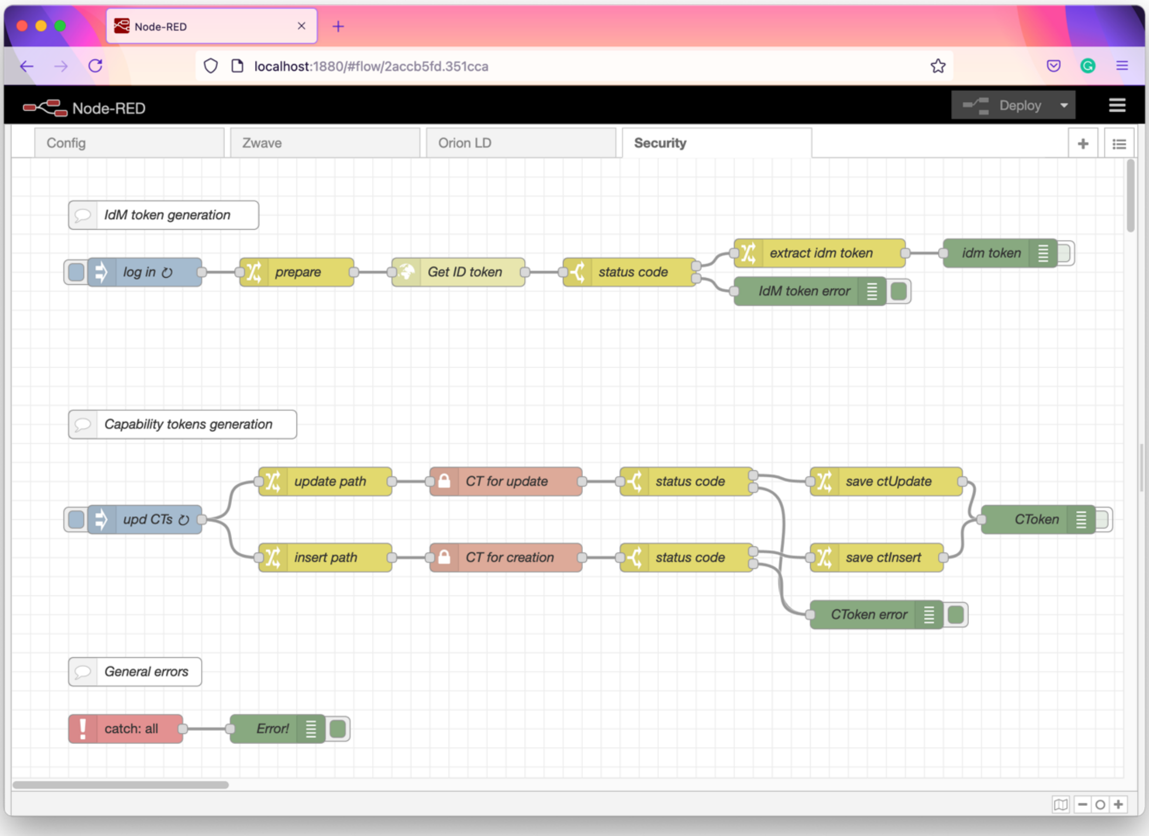


Figure 23: Z-wave Gateway - Security Flow (Node Red)

7 Conclusions

This document has described the work carried out so far within WP4 towards the development and implementation of the PHOENIX Smartness Hub. It currently focuses on the PoC site in Murcia, Spain, while the next versions will be extended to the other pilots. The document has provided a brief description of the PoC site and the data that is being gathered. It also presented the data models and storages being used to realize the edge KG, namely Smart Data Models and NGSI-LD brokers. Regarding the cloud KG we have identified relevant building models that can be reused, and we have presented a novel SRI semantic model for supporting SRI building's assessments and enabling automatic SRI computation. The document also presented the current status of the other components in the PHOENIX' functional layer, namely the user and grid centric analytical engines. For the user centric engine, we are currently focusing on optimizing user comfort by combining data from multiple sources. For the grid centric engine, we have developed a simulated micro grid, in order to test the algorithms being developed. Finally, the document concluded with a report on the progress of the Security & Privacy framework, which is part of the PHOENIX Smartness Hub. We have presented the risk analysis performed at the early stage of the project, and the status of the features being implemented.

8 Bibliography

Ferraiolo, D., Cugini, J., & Kuhn, D. R. (1995, December). Role-based access control (RBAC): Features and motivations. In Proceedings of 11th annual computer security application conference (pp. 241-48).

Yuan, E., & Tong, J. (2005, July). Attributed based access control (ABAC) for web services. In Web Services, 2005. ICWS 2005. Proceedings. 2005 IEEE International Conference on. IEEE.

Hernández-Ramos, J.L., A. J. Jara, L. Marín, and A. F. Skarmeta, "Dcapbac: Embedding authorization logic into smart things through ecc optimizations," International Journal of Computer Mathematics, no. just-accepted, pp. 1–22, 2014.

Boneh, D. and M. Franklin, "Identity-based encryption from the weil pairing," in Advances in Cryptology—CRYPTO 2001. Springer, 2001, pp. 213–229

Specification of the Identity Mixer Cryptographic Library. Version 2.3.40 IBM Research, Zurich January 30, 2013. Available http://www.zurich.ibm.com/idemix/

U-Prove Cryptographic Specification V1.1. Revision 3. Microsoft Corporation Authors: Christian Paquin, Greg Zaverucha, December 2013. Available at http://research.microsoft.com/en-us/projects/u-prove/

Heer, T., Garcia-Morchon, O., Hummen, R., Keoh, S. L., Kumar, S. S., & Wehrle, K. (2011). Security Challenges in the IP-based Internet of Things. Wireless Personal Communications, 61(3), 527-542.

Gusmeroli, S., Piccione, S., &Rotondi, D. (2013). A capability-based security approach to manage access control in the internet of things. Mathematical and Computer Modelling, 58(5), 1189-1205.

Hernández-Ramos, J. L., Jara, A. J., Marın, L., &Skarmeta, A. F. (2013). Distributed capability-based access control for the internet of things. Journal of Internet Services and Information Security (JISIS), 3(3/4), 1-16.

Hardy, N. "The Confused Deputy:(or why capabilities might have been invented)". ACM SIGOPS Operating Systems Review, 22(4), 36-38. 1988.

Gusmeroli, S., Piccione, S., &Rotondi, D. (2013). A capability-based security approach to manage access control in the internet of things. Mathematical and Computer Modelling, 58(5), 1189-1205. The application/json Media Type for JavaScript Object Notation (JSON) - https://tools.ietf.org/html/rfc4627

García, Á. L. (2019). DEEPaaS API: A REST API for machine learning and deep learning models. Journal of Open Source Software, 4(42), 1517.