Self Assessment Towards Optimization of Building Energy

Deliverable 1.4

Description of the system architecture of the SATO platform

Deliverable Lead: FC.ID

Deliverable due date: 31/07/2021

Actual submission date: 29/07/2021

Version: 2.1

This project receives funding in the European Commission's Horizon 2020 Research Programme under Grant Agreement Number 957128.

Document Control Page	
Title	Description of the system architecture of the SATO platform
Editor	FC.ID
Description	Describes the SATO platform architecture
Contributors	AAU, EKAG
Creation date	20/05/2021
Type	Report
Language	English
Audience	☐ public ☒ confidential
Review status	☐ Draft ☐ WP leader accepted ☒ Coordinator accepted
Action requested	☐ to be revised by Partners ☐ for approval by the WP leader ☐ for approval by the Project Coordinator ☐ for acknowledgement by Partners

Table of Contents

List of Figures

Revision history

Version	Author(s)	Changes	Date
1.0	Vinícius Cogo, José Cecílio, Ana Paula Afonso, Pedro Ferreira, Alexandre Nascimento, André Gil, Vasco Ferreira (FC.ID), Thomas Fehr (EKAG)	First document draft	15/07/2021
2.0	Thomas Fehr (EKAG)	First review	21/07/2021
2.1	Pedro Ferreira	Final check, formatting, and minor corrections before submission	29/07/2021

DISCLAIMER

The sole responsibility for the content of this publication lies with the SATO project and in no way reflects the views of the European Union.

EXECUTIVE SUMMARY / ABSTRACT / SCOPE

The SATO platform is a complete solution that encompasses communication, computing, storage, and energy management service components, supporting a self-assessment framework and several energy management self-optimization services. These services take advantage of building data and assessments to increase energy flexibility, efficiency, and user satisfaction.

Deliverable 1.4 (D1.4) presents the overall conceptual architecture of the SATO platform. The main outcome of this document is the description of the six architectural blocks that compose the SATO platform: external data sources, buildings, SATO middleware, self-assessment framework, self-optimization services, and actors. Altogether, these blocks include more than 30 components with specific roles that are also described together with many foreseen data flows.

D1.4 is a direct output of Task 1.3 (Requirements and System Architecture for the SATO platform and for it to support SRI calculation) from WP1 (Specification and Requirements for SATO). It will provide guidance for the ongoing development in WP2 (Development of integrated technical Platform for SATO) and will guide the specification and integration of concrete internal components associated with WP3 (Development of SATO self-assessment framework) and WP4 (Development of SATO self-assessment and self-optimization services toolbox).

1. Introduction

1.1. Motivation

One of the SATO project pillars is the creation of a new energy self-assessment and optimization platform – the SATO platform – able to integrate all energy-related Equipment and Building Components (EBCs). This platform will support a Self-Assessment Framework (SAF) aligned with the structure of the Smart Readiness Indicator (SRI), a BIM-based interface for analysis and visualization of assessments, and the development of energy management services for increased energy flexibility, efficiency, and user satisfaction.

1.2. Objective

The objective of this deliverable is to describe the system architecture of the SATO platform, its components, and their integration. Although the main objective is to describe the platform system architecture, the initial implementation of components is also briefly described (if already available).

D1.4 is a direct output of Task T1.3 (Requirements and System Architecture for the SATO platform and for it to support SRI calculation) within WP1 (Specification and Requirements for SATO), to guide the development of the platform in WP2 (Development of integrated technical Platform for SATO).

1.3. Structure of the document

The remaining of this document is divided into ten sections. Section 2 presents an overview of the SATO architecture and introduces the main blocks of components foreseen for the SATO platform. Sections 3-10 describe in detail the components within each block and their integration in the whole data flow. Finally, Section 11 concludes this document by guiding the subsequent steps related to the SATO platform's development and deployment.

2. Overview of the SATO architecture

The SATO Platform comprises six blocks of components and players: external data sources, buildings, SATO middleware, self-assessment framework, self-optimization services, and actors. They are depicted in Figure 1 and overviewed in this section.

Each mentioned block is composed of several internal system components, each with specific features and a well-defined role. The current main components in each block are related to high-level abstractions of roles. In the future, as the components become developed in other WPs, they might be subdivided, thus originating additional, more specific components. Since this will likely happen with the self-assessment services, self-optimization services, and assessments components, updated versions of the architecture will be released in Deliverables 2.2 and 2.5. In this section, we briefly introduce the six blocks of the SATO architecture. Each of them will be described in more detail in subsequent sections.

Buildings are the main target object for the energy assessment and optimization services of the SATO project. They are the physical space where the devices, that will interact with the platform, will be deployed. We foresee three main strategies for devices providing data to the SATO platform and receiving actuations: directly, through a gateway, and through a cloud-based service. These three scenarios cover most of the requirements of external commercial platforms and SATO partners. Additionally, the data provision can be implemented either proactively through producer-consumer mechanisms that directly connect buildings (and forward data from devices) to the SATO platform, or reactively through the SATO platform deploying pooling components that fetch data from devices in buildings and send it to the appropriate internal components, sparking the normal data flow. More details on the components from this block are described in Section 3.

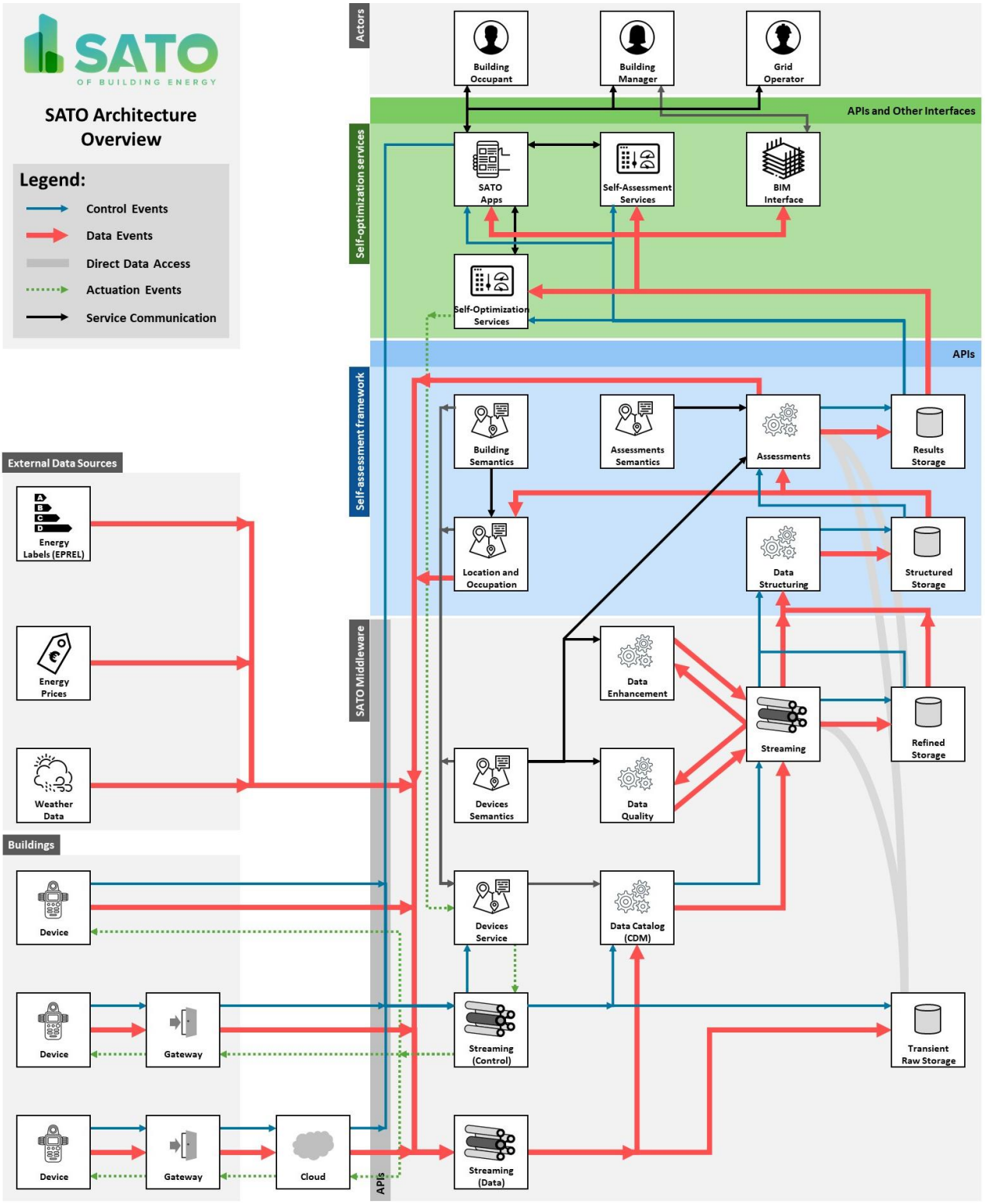

Figure 1 - Overview of the SATO architecture.

External data sources are databases that provide context and environment information for the SATO platform. They are read-only data that will be forwarded to the SATO platform. Examples of foreseen external sources include (but are not limited to) energy labeling of appliances (e.g., EPREL), weather data, and energy prices. More details about them can be seen in Section 4.

SATO middleware is the core block from the SATO architecture in terms of ingesting data, organizing it, and preparing it to be consumed by the main processing tasks of the self-assessment framework and optimization services. This block is subdivided into event streaming components that will bridge buildings and processing components, services that standardize events being ingested by the platform, services for keeping up-to-date snapshots of devices on each building (and providing semantically coherent data about them), and data enhancing components that will assess and improve the quality of data being provided by the platform. All components in this large block are described in Sections 5-8.

The SAF is the main processing block of the platform, both in terms of quantity and variety of data. It will use the data ingested by the SATO platform through data analysis and machine learning to build the assessments that will report on the energy performance of buildings and EBCs, on building occupancy, on equipment faults, and on the SRI. This block provides a series of assessment results that complement the available building data, serving as the inputs for the self-optimization services block that will make energy management decisions. Sections 9 and 10 present more detailed descriptions of the SAF and self-optimization services, respectively.

Finally, the *Actors* block represents the users of the SATO platform. It includes building occupants/owners, building managers, and grid operators, which are humans that will directly benefit from the SATO project (and its platform) through assessments and optimized energy management decisions, improved comfort, or more flexible infrastructures than the ones currently available.

Arrows in Figure 1 depict data flows of different event types that will become clearer with the descriptions provided. They include control events (blue lines) related to the status of the monitored buildings and of the platform, data events (red lines) that provide most of the data and building measurements that will be ingested and processed by the SATO platform, and actuation events (green dashed lines) that represent actions and control commands taken by the self-optimization services and managers to change the behavior of buildings. Other interactions like direct data access (thick light-grey lines) and platform operational service communication (thin black lines) are also shown in the architecture of the SATO platform presented.

We now pivot our focus to zooming into the architecture of the SATO platform and describing each block of components from Figure 1 in the subsequent sections.

3. Buildings and their devices

The SATO platform architecture is designed to deal with incoming data and outgoing actuation/control commands. It offers support for different types of integration with devices, including: i) direct communication between devices and the platform; ii) communication between a building gateway and the platform, and/or iii) communication between cloud services offered by third parties (e.g., device suppliers or other platforms) and the platform.

By default, the proposed architecture is designed to handle data streams, but since some devices and platforms are not stream-oriented, the SATO architecture follows a modular solution where specific adapters can handle distinct device requirements. Adapters abstract the communication from non-stream devices by receiving their data events and injecting them into the platform streaming components. From that point on, those data will follow the same procedures of streamed events. When a new device or platform is integrated using specific adapters, an initial registration step is needed. The registration establishes the user/device credentials, useful metadata, and initial configurations. In contrast, when the streaming capability is available at the device or platform, the registration information can be directly injected into the stream module. The SATO platform will then consume the data, perform the registration process, and generate all the internal information for the device data mapping. Upon registration, an internal SATO ID is generated and the first communication with the device is established. The internal ID is used to map all data (events or commands) of that device, thereby abstracting any underlying IDs and standardizing the identification and representation of all devices within the platform.

In either case, stream-oriented or not, after the initial device configuration, the SATO platform is ready to receive data or to send commands to devices.

When platform services generate device actuation or control command events, the same problem arises. If platforms and devices can consume those events with a streaming component, they are streamed to those devices. Otherwise, those events are consumed by the appropriate adapter, which will follow the device requirements and send them through the device-specific interface or protocol.

Considering data events, the SATO platform is designed to deal with different data formats used by devices. For that, it uses a common data model and a specific component, allowing the conversion of specific device data formats to a unique SATO platform format. That component is described in Section 7.

4. External data sources

Some components of the self-assessment and self-optimization blocks will require data from external sources to implement and operationalize assessments and services. In general, various assessments to be developed in WP3 and energy efficiency and flexibility optimization services to be developed in WP4 will use historical and forecast weather data or energy price data. Additionally, the assessments related to the real-life energy performance of appliances will be compared to data provided by market suppliers stored in the European Product Database for Energy Labelling (EPREL).

The SATO platform includes components to integrate the necessary external data and make it available for the development and operation of assessments and services. The considered external data sources will be described in the following subsections.

4.1. EPREL

4.1.1. EPREL information

The European Product Database for Energy Labelling (EPREL) [1] is an online product registration database managed by the European Commission (EC) and contains energy efficiency information about the devices covered by the Energy Labelling Regulation (EU) 2017/1369 [2]. This regulation replaces the Directive 2010/30/EU, maintaining essentially the same scope but updating the energy labeling framework considering the technological progress for energy efficiency in products achieved over recent years. Since January 1st, 2019, all EU suppliers are legally required to register in the EPREL database new product models that the Energy Labelling covers before they are placed on the EU market for the first time.

The main purposes of the EPREL database are, among others: to provide the public with information about products placed on the EU market and their energy labels; and to provide the EC with up-to-date information on products' energy efficiency for reviewing energy labels [1].

Energy labels

The purpose of EU energy labels is to indicate the energy efficiency of products, enabling consumers to make informed choices on more energy-efficient products while also reducing greenhouse gas emissions. When a supplier places an energy-consuming product on the market, each unit should be accompanied by a label in paper format that should be easily recognizable. The energy label contains information separated into at least four sections (Figure 2):

(1) Product details: specific information about the device, e.g., the brand and model.

(2) Energy class: a green to red color scale associated to a letter, from A (most efficient) to G (least efficient) that gives an idea of the product's energy consumption. In the old label the class is on a scale from A^{+++} to D, while in the rescaled labels scale is from A to G (Figure 2).

(3) Consumption, efficiency, capacity, or other information depending on the product type.

(4) QR code: placed on the top right corner in the rescaled energy labels, that gives access to additional information about the product.

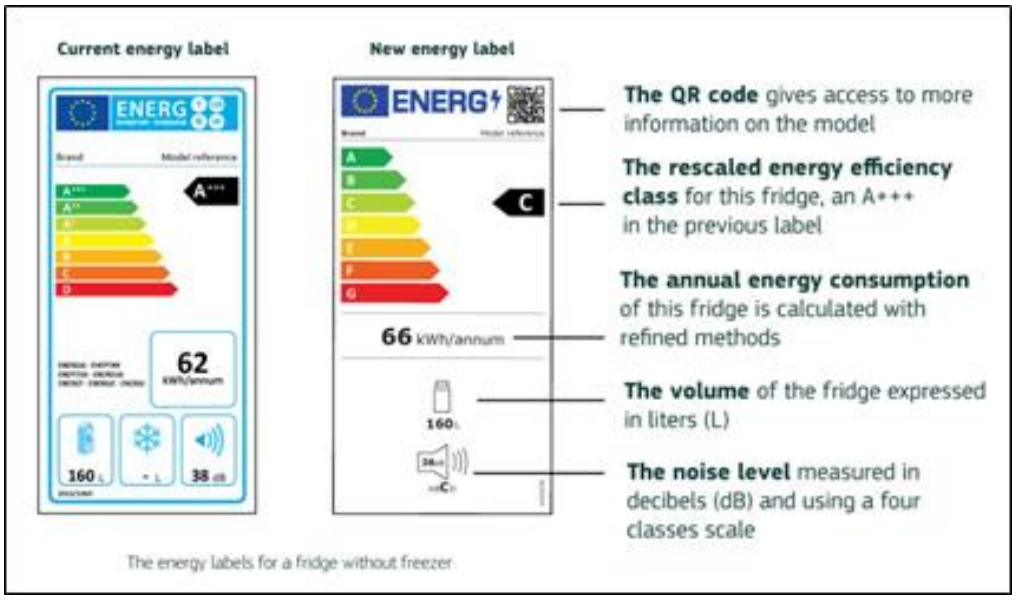

Figure 2 - Example of the past energy label (left) and new label (right) for a fridge [3].

Data available in EPREL

The EPREL database consists of a public part, a compliance part, and an online portal [4]. The portal gives access to the two parts, the public one allowing to consult the previously entered information by the suppliers of products and the EC. In the compliance part, it is possible to register and manage an organization, register a device, and enter information related to the energy label, technical documentation, and compliance monitoring [1].

The public information available for each device depends on its type, however there is some information that is common to all, namely [1]:

- Name, address, and contact details of the supplier.
- Energy label in electronic format.
- Energy efficiency class and other parameters of the label.

The supplier also enters some information that is not publicly available, but is necessary for the market surveillance authorities:

- A general description of the model.
- References to the harmonized standards applied.
- Specific precautions to be taken when the model is assembled or installed.
- Measured technical parameters of the model.
- Calculations performed with the measured parameters.

Additionally, the EPREL system associates to each device, some important information:

- Model registration number, used to uniquely identify the device in the database. It is also associated with the QR code on the device energy label.
- Product group code, used to identify the device type. Currently, the product group codes supported are:
    - HOUSEHOLD_DISHWASHER_2019

- HOUSEHOLD_WASHING_MACHINE_2019
- HOUSEHOLD_WASHER_DRIER_2019
- ELETRONIC_DISPLAY
- HOUSEHOLD_REFRIGERATING_APPLIANCE_2019

respectively, for dishwashers, washing machines, washer driers, electronic displays, and refrigerator devices. Lamps are planned to be added on September 1st, 2021, and other device types in the following years. This means that 5 product groups will be rescaled in 2021.

It is important to note that the measurement units for each metric are not stored on EPREL, but they are available on the labeling documentation. For example, the measurement units for electronic displays are available on the Regulation on energy labeling for electronic displays (EU) 2019/2013 [5]. These measurement units must be used by all suppliers when inserting device information in EPREL.

For the current supported device types, the information and metrics registered in the EPREL database are [6]:

- **Electronic displays:**
  - Energy class (A, B, C, D, E, F, G)
  - On mode power demand (Watts)
  - Off mode power demand (Watts)
  - Standby mode power demand (Watts)
  - Networked standby mode power demand (Watts)
  - Visible screen area (Dm^2)
  - Category (Television, Monitor, Signage, Other)
  - Panel technology (LCD, LED LCD, QLED LCD, OLED, MicroLED, QLED, FED, EPD, Other)
  - Size ratio x
  - Size ratio y
  - Resolution horizontal (Pixels)
  - Resolution vertical (Pixels)
  - Screen diagonal (cm)
- **Dishwashers:**
  - Energy class (A, B, C, D, E, F, G)
  - Energy efficiency index
  - Rated capacity (place settings, a set of tableware for one person)
  - Energy consumption [per cycle, based on the eco programme] (Kilowatts)
  - Energy consumption [per 100 cycles, based on the eco programme] (Kilowatts)
  - Water consumption [per cycle, based on the eco programme] (Liters)
  - Programme duration for the eco programme (Hour:Minutes)
  - Off mode power demand (Watts)
  - Standby mode power demand (Watts)
  - Networked standby mode power demand (Watts)
  - Delay start mode power demand (Watts)
  - Noise emission class for the eco programme (A, B, C, D)
  - Noise emissions for the eco programme (Decibel)
  - Height (cm)

- Width (cm)
- Depth (cm)
- Cleaning performance index
- Drying performance index

- **Refrigerator devices:**
  - Energy class (A, B, C, D, E, F, G)
  - Energy efficiency index
  - Annual energy consumption (Kilowatts)
  - Noise emission class (A, B, C, D)
  - Noise emissions (Decibel)
  - Low noise appliance (Yes, No)
  - Total volume (Liters or dm^3)
  - Height (mm)
  - Width (mm)
  - Depth (mm)
  - Design type (Built in, Free standing)
  - Compartment:
    - Compartment volume (Liters or dm^3)
    - Compartment type (Pantry, Wine storage, Cellar, Fresh food, Chill, Zero star, One star, Two star, Three star, Four star, Two star section, Variable temp)
- **Washing machines:**
  - Energy class (A, B, C, D, E, F, G)
  - Energy efficiency index
  - Rated capacity (Kg)
  - Energy consumption [per cycle, eco 40-60 programme] (kilowatt-hour)
  - Weighted energy consumption [per 100 cycles, eco 40-60 programme] (kilowatt-hour)
  - Water consumption [per cycle, eco 40-60 programme] (Liters)
  - Washing efficiency index
  - Rinsing effectiveness (g/kg)
  - Maximum temperature inside the treated textile [Rated capacity] (Celsius)
  - Maximum temperature inside the treated textile [Half capacity] (Celsius)
  - Maximum temperature inside the treated textile [Quarter capacity] (Celsius)
  - Remaining moisture [Rated capacity] (%)
  - Remaining moisture [Half capacity] (%)
  - Remaining moisture [Quarter capacity] (%)
  - Spin-drying efficiency class (A, B, C, D, E, F, G)
  - Spin speed [Rated capacity] (rpm)
  - Spin speed [Half capacity] (rpm)
  - Spin speed [Quarter capacity] (rpm)
  - Programme duration [Rated capacity] (Hour:Minutes)
  - Programme duration [Half capacity] (Hour:Minutes)
  - Programme duration [Quarter capacity] Hour:Minutes)

- Off mode power consumption (Watts)
- Standby mode power consumption (Watts)
- Networked standby power consumption (Watts)
- Delay start power consumption (Watts)
- Noise emissions class (A, B, C, D, E, F, G)
- Noise emissions (Decibel)
- Height (cm)
- Width (cm)
- Depth (cm)

- **Washer drier machines:**
  - Energy class for complete/washing cycle (A, B, C, D, E, F, G)
  - Energy efficiency index for complete/washing cycle
  - Rated capacity for complete/washing cycle (Kg)
  - Energy consumption for complete/washing [per cycle, eco 40-60 programme] (kilowatt-hour)
  - Weighted energy consumption for complete/washing [per 100 cycles, eco 40-60 programme] (kilowatt-hour)
  - Water consumption for complete/washing [per cycle, eco 40-60 programme] (Liters)
  - Washing efficiency index for complete/washing cycle
  - Rinsing effectiveness for complete/washing cycle (g/kg)
  - Maximum temperature inside the treated textile for complete/washing cycle [Rated capacity] (Celsius)
  - Maximum temperature inside the treated textile for complete/washing cycle [Half capacity] (Celsius)
  - Maximum temperature inside the treated textile for complete/washing cycle [Quarter capacity] (Celsius)
  - Remaining moisture [Rated capacity] (%)
  - Remaining moisture [Half capacity] (%)
  - Remaining moisture [Quarter capacity] (%)
  - Spin-drying efficiency class (A, B, C, D, E, F, G)
  - Spin speed [Rated capacity] (rpm)
  - Spin speed [Half capacity] (rpm)
  - Spin speed [Quarter capacity] (rpm)
  - Programme duration for complete/washing cycle [Rated capacity] (Hour:Minutes)
  - Programme duration for complete/washing cycle [Half capacity] (Hour:Minutes)
  - Programme duration for complete/washing cycle [Quarter capacity] (Hour:Minutes)
  - Off mode power consumption (Watts)
  - Standby mode power consumption (Watts)
  - Networked standby power consumption (Watts)
  - Delay start power consumption (Watts)
  - Noise emissions class (A, B, C, D, E, F, G)
  - Noise emissions (Decibel)

- Height (cm)
- Width (cm)
- Depth (cm)

4.1.2. Integration with SATO platform

Data stored in EPREL can be accessed either by scanning the QR code on the energy label, which redirects to a web page with all the information, or by resorting to the EPREL REST API [7], which is the solution used for the integration with the SATO platform. The API provides five request calls:

- **Get list of products groups -** Request used to get full list of product group codes.
- **Get product group's models -** Request used to search for all the models within a group. The mandatory parameter needed is the product group code. This request is the only one that is restricted, since it needs an API key to be used. The request for this key is not yet available, and it is not known when it will be available.
- **Get product group's model by registration number -** Request used to get the data for a single model in a product group. The mandatory parameters are the product group code and the model registration number.
- **Get product group's model label by registration number -** Request used to get the label of a single model in a product group. The mandatory parameters are the product group code and the model registration number.
- **Get product group's model fiche by registration number -** Request used to get the fiche (product information sheet) of a single model in a product group. The mandatory parameters are the product group code and the model registration number.

The data fetched from the EPREL database will be used by SATO real-life performance assessment of appliances (WP3) as a reference for comparison to the real-life metrics that will be calculated using sensor-measured data. Additionally, some of the metadata of the device model is needed for the calculation of some metrics. For example, in the case of the electronic displays, the visible screen area is used to obtain the energy class. That measure is part of the metadata stored on the EPREL database.

The data stored on the EPREL database is fetched by using the request call `Get product group's model by registration number`. The call must have the following format:

```
https://eprel.ec.europa.eu/api/products/[PRODUCT_GROUP]/[REGISTRATION_NUMBER]
```

The `PRODUCT_GROUP` *field* must be a valid product group code. The `REGISTRATION_NUMBER` field is the model registration number used in the EPREL database as a unique identifier for the model [6]. To obtain this number it is necessary to scan the QR code on the energy label, which redirects to the EPREL public website that gives the registration number of the model in the corresponding URL.

For example, for the case of a washer drier energy label (right part of Figure 3), the URL for the model page is https://eprel.ec.europa.eu/screen/product/washerdriers2019/300268 and the registration number is 300268.

The data from EPREL will be fetched when a device is first registered in the SATO platform, by using the previously described call. The output of the call is shown in Figure 3.

General information

Overall dimensions	85 (Height) x 60 (Width) x 58 (Depth)	cm
Type	Free-standing appliance	
Off-mode	0,5	W
Standby mode	1	W
Delay start	4	W
Networked standby	2	W
Releases silver ions	No	
Minimum duration of the guarantee offered by the supplier	24	months
Additional information	-	

Weblink to the supplier's website
https://corporate.haier-europe.com/en/

	D	A	
Energy consumption per cycle	3,078	0,494	kWh
Weighted energy consumption [per 100 cycles]	308	49	kWh
Water consumption	86	49	litre
Maximum temperature inside the treated textile (Rated washing capacity)	26	37	°C
Maximum temperature inside the treated textile (Half)	22	30	°C
Maximum temperature inside the treated textile (Quarter)	-	24	°C
Washing efficiency index	1,031	1,031	
Rinsing effectiveness	4,9	4,9	g/kg dry textile
Weighted remaining moisture content		53,9	%
Spin speed (Rated washing capacity)		1 351	rpm
Spin speed (Half)		1 351	rpm
Spin speed (Quarter)		1 351	rpm
Spin-drying efficiency class		B	(A - G)
Eco 40-60 programme duration (Rated washing capacity)		3:48	(h:min)
Eco 40-60 programme duration (Half)		2:54	(h:min)
Eco 40-60 programme duration (Quarter)		2:54	(h:min)
Wash and dry cycle duration (Rated capacity)	9:30		(h:min)
Wash and dry cycle duration (Half)	5:30		(h:min)
Airborne acoustical noise emissions during the spinning phase for the eco 40-60 washing cycle at rated washing capacity		78	dB(A) re 1 pW
Airborne acoustical noise emission class for the spinning phase for the eco 40-60 programme at rated washing capacity		C	(A - D)
Airborne acoustical noise emissions during the spinning phase for the eco 40-60 washing cycle at rated washing capacity		78	dB(A) re 1 pW
Airborne acoustical noise emission class for the spinning phase for the eco 40-60 programme at rated washing capacity		C	(A - D)

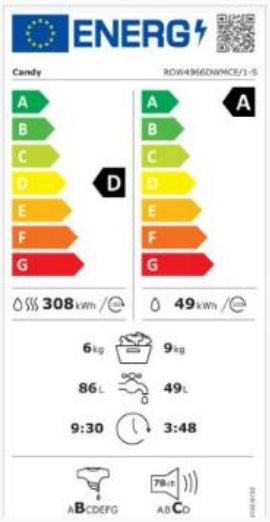

Figure 3 - Example output of an EPREL database API call. Left: assessment data of the washer drier. Right: the corresponding EU energy label.

4.2. Weather data

Various assessments to be developed in WP3 and energy efficiency and flexibility optimization services to be developed in WP4 require access to historical and forecast weather data. There are three main options to obtain such data for a given location: local weather station and predictive models, regional or national weather data services, or global weather data services. Since the SATO platform aims to be a scalable and accessible platform, the only viable option are global weather data services. Although possible, integrating local weather stations would require dealing with the heterogeneity of weather stations across building locations and having largely increased computational requirements to handle the corresponding predictive models. Regional or national weather services also impose the heterogeneity limitation to integrate with the SATO platform but have yet another limitation: usually they provide data only for certain locations, commonly the main regional or national cities. Therefore, the generalized solution is to resort to globalized weather data services that will provide historical and forecast data for any given location on Earth.

Considering global weather data services, there are many options available to choose from. To select an appropriate one, several important features should be considered: cost, availability, scalability, coverage, and integration mechanism. The following options were evaluated:

- AccuWeather[1]
- OpenWeather[2]
- Tomorrow.io[3]
- WeatherAPI.com[4]

After considering the options and corresponding features, OpenWeather (https://openweathermap.org/) was selected since it provides a suitable compromise between features. The service provides current, historical, and forecast weather data on any location specified by latitude and longitude coordinates. To generate the forecasts, OpenWeather numerical prediction uses convolutional neural networks to combine many sources of information. The following data options are available through a web API:

- Current weather
- Minute forecast for 1 hour
- Hourly forecast for 48 hours
- Daily forecast for 7 days
- National weather alerts
- Historical weather data for the previous 5 days

A free plan is available, allowing for 60 API calls per minute on the current weather and 1000 calls per day on the remaining elements. These numbers of API calls are adequate for the SATO project, considering the 8 pilots, permitting different forecast horizons and granularities to be used, e.g.:

- Short-term forecast: one-minute forecast for one hour, updated every 15 minutes, resulting in 768 calls per day (8 pilots, 24 hours, 4 calls per hour per pilot).
- Mid-term forecast: one-hour forecast for 48 hours, updated every hour, resulting in 192 calls per day (8 pilots, 24 hours, 1 call per hour per pilot)

[1] https://developer.accuweather.com/
[2] https://openweathermap.org/api
[3] https://www.tomorrow.io/weather-api/
[4] https://www.weatherapi.com/docs/

For a scaled SATO platform derived business, a paid plan would have to be purchased, requiring a more in-depth cost/benefit analysis.

The integration with the SATO platform will be done using the web API provided by the service. A component in the SATO platform, using a specific adapter, will be responsible to fetch the data at the required time instants and insert it in the platform streaming component, so that the forecast becomes stored and ready for use. As an example, consider the following API call that fetches the current weather, a one-hour forecast for 48 hours, and 7-days daily forecast for a location in Lisbon, Portugal:

```
https://api.openweathermap.org/data/2.5/onecall?lat=38.7167&lon=-
9.1333&exclude=minutely,alerts&appid={omitted key value}&units=metric
```

The result would be the following data formatted in JSON language:


```
{
"lat":38.7167,"lon":-9.1333,
"timezone":"Europe/Lisbon","timezone_offset":3600,

"current":

{"dt":1626259753,"sunrise":1626240189,"sunset":1626292874,"temp":25.25,"feels_like":25.27,"pressu
re":1017,"humidity":55,"dew_point":15.57,"uvi":8.08,"clouds":0,"visibility":10000,"wind_speed":3.
6,"wind_deg":20,"weather":[{"id":800,"main":"Clear","description":"clear sky","icon":"01d"}]},

"hourly":[

{"dt":1626256800,"temp":24.88,"feels_like":24.83,"pressure":1017,"humidity":54,"dew_point":14.94,
"uvi":5.98,"clouds":0,"visibility":10000,"wind_speed":2.12,"wind_deg":3,"wind_gust":3.38,"weather
":[{"id":800,"main":"Clear","description":"clear sky","icon":"01d"}],"pop":0},

{"dt":1626260400,"temp":25.25,"feels_like":25.27,"pressure":1017,"humidity":55,"dew_point":15.57,
"uvi":8.08,"clouds":0,"visibility":10000,"wind_speed":2.36,"wind_deg":11,"wind_gust":3.78,"weathe
r":[{"id":800,"main":"Clear","description":"clear sky","icon":"01d"}],"pop":0},
```


… (*These blocks of information repeat for every hour within the 48 hours predictive horizon*) …


```
{"dt":1626426000,"temp":24.7,"feels_like":24.4,"pressure":1015,"humidity":45,"dew_point":10.85,"u
vi":3.65,"clouds":0,"visibility":10000,"wind_speed":1.19,"wind_deg":129,"wind_gust":1.34,"weather
":[{"id":800,"main":"Clear","description":"clear sky","icon":"01d"}],"pop":0}
],

"daily":[

{"dt":1626264000,"sunrise":1626240189,"sunset":1626292874,"moonrise":1626255540,"moonset":0,"moon
_phase":0.15,"temp":{"day":25.67,"min":15.5,"max":28.73,"night":20.85,"eve":25.53,"morn":15.62},"
feels_like":{"day":25.65,"night":20.61,"eve":25.34,"morn":15.46},"pressure":1017,"humidity":52,"d
ew_point":15.09,"wind_speed":7.92,"wind_deg":326,"wind_gust":13.9,"weather":[{"id":800,"main":"Cl
ear","description":"clear sky","icon":"01d"}],"clouds":0,"pop":0,"uvi":9.42},

{"dt":1626350400,"sunrise":1626326633,"sunset":1626379243,"moonrise":1626345960,"moonset":1626303
960,"moon_phase":0.18,"temp":{"day":28.44,"min":19.74,"max":28.92,"night":22.19,"eve":25.85,"morn
":19.96},"feels_like":{"day":27.59,"night":21.82,"eve":25.59,"morn":19.6},"pressure":1014,"humidi
ty":33,"dew_point":9.85,"wind_speed":6.13,"wind_deg":239,"wind_gust":7.61,"weather":[{"id":801,"m
ain":"Clouds","description":"few clouds","icon":"02d"}],"clouds":19,"pop":0,"uvi":9.57},
```


… (*These blocks of information repeat for every day in the 7 days predictive horizon*) …


```
{"dt":1626868800,"sunrise":1626845310,"sunset":1626897421,"moonrise":1626889980,"moonset":1626834
240,"moon_phase":0.4,"temp":{"day":24.55,"min":16.95,"max":24.57,"night":19.77,"eve":22.63,"morn"
:16.95},"feels_like":{"day":24.24,"night":19.97,"eve":22.59,"morn":17.05},"pressure":1016,"humidi
ty":45,"dew_point":11.53,"wind_speed":5.6,"wind_deg":281,"wind_gust":8.07,"weather":[{"id":800,"m
ain":"Clear","description":"clear sky","icon":"01d"}],"clouds":0,"pop":0,"uvi":7}
]
}
```

4.3. Energy prices and CO_2 data

Electricity prices are important in the context of flexibility management and demand-side response methodologies. Two cases can be considered: a daily/weekly static price plan or a dynamic price plan where the price varies with time. These data are provided locally (referring to the electricity consumption location) by the electricity reseller.

Regarding the dynamic price case, the data provision mechanism will vary locally depending on the electricity reseller and the demand-side response methodology. In this case, a data integration procedure must be implemented, either through a streaming approach or via an adapter (like the use of weather data APIs). In cases where the price is given by a static daily or weekly pattern, the plan can simply be stored in the platform data storage.

At the time of writing this document, it is not completely clear how and where the demand response and flexibility management services will be demonstrated in the SATO project. Therefore, specific approaches cannot yet be described.

Historical, current, and forecast CO_2 emission data can also be integrated if deemed useful for SA&O, reporting, or user engagement purposes.

For instance, electicityMap (https://www.electricitymap.org/map) provides electricity prices, electricity consumption, and CO_2 emissions for European regions in an hourly basis, with integration through an easy-to-use API.

5. Data ingestion

As depicted in the detail presented in Figure 4, the first contact points between buildings and the SATO platform are two data streaming instances. One of them receives all control events (e.g., register a new device), while the other receives data events (e.g., measurements).

Devices in buildings produce events (control or data) that are written to the appropriate channel of the streaming instances. These channels (or topics) follow a producer/consumer paradigm receiving events (using one of the three strategies described in Section 3), and simply forwarding them to registered consumers.

In the case of the streaming for control events, these are forwarded to the Devices Services (described in Section 6) and to the Transient Storage (explained below) as soon as possible. They are also sent to the Data Catalog component (described in Section 7), which will standardize the data and send them to a second-level streaming (Section 8.1).

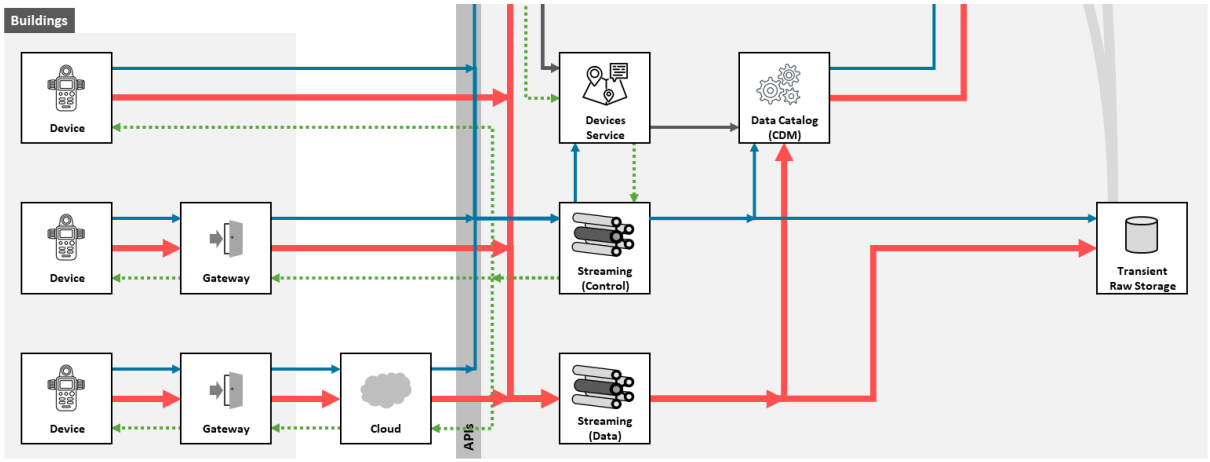

Figure 4 - Detail of the SATO platform architecture: buildings and the first part of the SATO middleware.

The streaming component for data events receives data from building devices and from external sources, and forwards them to the Data Catalog and Transient Storage components. The data streaming component is unidirectional since buildings only produce data (i.e., they do not consume data from one another).

The separation between control and data events (and their respective instances) enables the SATO platform to appropriately schedule processing and communication resources following priority policies. For instance, we may assume that control events are more important than data events, since the former includes commands, configurations, and possibly many other sporadic information that may interfere with the platform and building functionality. Data events may have lower priority because they present a voluminous data flow with little influence in the overall building and platform functionality, and they do not have strict hard real-time requirements. Moreover, many times data can be recovered or inferred by data quality components in case of a temporary failure. This scheduling prioritization enables the SATO platform to implement different modes of operation, allocating more resources for the control streaming when the system is under stress or peak loads, or allocating balanced resources for both streaming instances when the workload is under certain thresholds.

The Transient Storage is a component responsible for consuming and ingesting information as fast as possible to reduce the chances of the SATO platform losing any control or data events. Considering the diversity of devices, gateways, and platforms that may interact with the SATO platform, this storage considers heterogeneous JSON events that are persistently stored in filesystems as a temporary buffer. The stored events are directly accessible to later processing components since, due to faults or timing issues, they may be lost within or unavailable to the remaining data flow.

Information is kept in Transient Storage only for a timeframe window. After the expected lifespan, the Transient Storage clears expired data. This way we guarantee that storage requirements do not grow indefinitely and allow this component to efficiently support the required throughput of the platform in an easier way. Alternatively, the Transient Storage may be configured to have a specific maximum size and store information while it has space. When the space limit is achieved, the Transient Storage may enter in a rotation working mode, which deletes old entries to open room for new ones. By storing information within a certain timeframe, we allow the possibility of some processing component increasing the probability of the event reaching the Refined Storage component. It may happen that the event (1) reached the Refined Storage through the normal data flow (i.e., through the Data Catalog component), (2) was detected as missing and was inferred by the Data Quality component, or (3) was directly collected from the Transient Storage and stored into the Refined Storage.

6. Devices services and semantics

Syntactic and semantic interoperability were some of the main requirements identified for the SATO platform in Deliverable D1.3. Device Services and the Device Semantics are two critical components of the SATO platform that will respectively address these two requirements. They offer services that enable the interaction with heterogeneous devices and commercial platforms.

Since the integration of heterogeneous IoT platforms and smart systems results in different interfaces, the optimization of energy resources becomes harder since potentially incompatible and different technologies and systems are used. The interoperable strategies proposed by the Devices Services component are required for solving this challenge.

The Device Services component provides a unifying REST interface to other components of the platform, abstracting the heterogeneity of all the integrated underlying platforms and smart systems. By doing so, the self-assessments and optimization services can more easily send actuations to the devices by only needing to be compatible with the interface of the Devices Services. Figure 5 shows the internal architecture of the Device Services component.

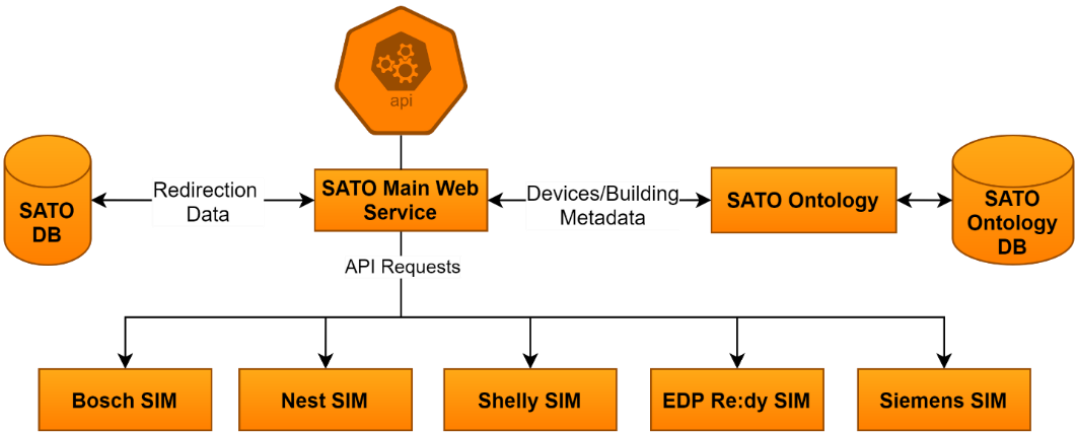

Figure 5 – Architecture of the device services component.

This component abstracts platform-specific characteristics by offering a global API that provides a unified way for upper layers to interact with the integrated platforms. The proposed architecture is supported by a generic structure called SATO Integration module (SIM), which enables integrating commercial platforms with the SATO platform in a standardized way by adapting it to each commercial platform. For instance, Figure 5 depicts the device services component with five SIM instances (Bosch, Nest, Shelly, EDP Re:dy, and Siemens). The generic architecture of this SIM is shown in Figure 6. This module must be implemented for each integrated platform and is responsible for the direct mapping between SATO API calls and proprietary platform calls.

When a request arrives at the Device services component, it is handled by the SATO API, identifying the corresponding SATO Integration module and forwards the request to it. Then, the SATO Integration module converts the request into specific requests according to the rules implemented and needed to exchange the request with the integrated platform.

The SATO Integration module follows a modular architecture that allows one to implement other mechanisms to interact with the integrated platform or device to deal with different platforms and vendors. Since some platforms or devices may use publish/subscribe mechanisms, this module can handle this type of communication. Additionally, and according to the platform, other communication mechanisms can be added to the module.

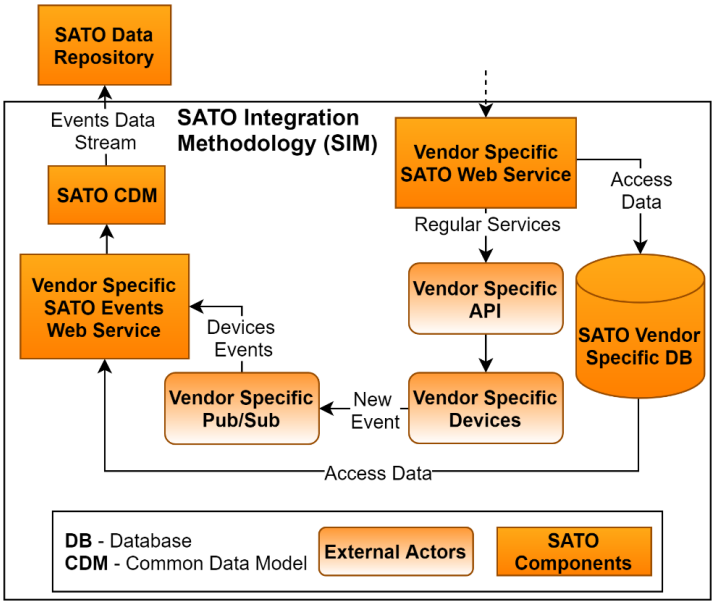

Figure 6 – SATO integration module.

Since the SATO Integration module is responsible for dealing with the platform that we want to integrate, it is also responsible for mapping all incoming data. The SATO architecture defines a common data model (CDM) that is used inside the SATO platform. To address this requirement, the SATO Integration module needs to convert proprietary data formats into SATO data format, with the SATO CDM component being responsible for this conversion and detailed in Section 7.

The Device Semantics component stores and provides information about devices, platforms, and other soft sensors (e.g., the EPREL external data source). Examples of information include the device's proprietary ID, manufacturer, the type of properties measured/controlled by the device, among others. Other components can request information about devices (e.g., their specific capabilities) with this mapping of properties, independently of the device's underlying representation. This device semantics comprises ontologies that persist data using semantic triples (e.g., an RDF triple [8] comprises the subject, predicate, and object). The set of these triples form a graph with all its data representing the context.

Defining an ontology for the SATO platform enables achieving standardization in integrating heterogeneous solutions. The Devices Semantics will reuse well-established ontologies about the devices domain and will extend them for better coverage. Figure 7 illustrates the devices ontology designed for the SATO project, which will store information about the devices in the Device Semantics component.

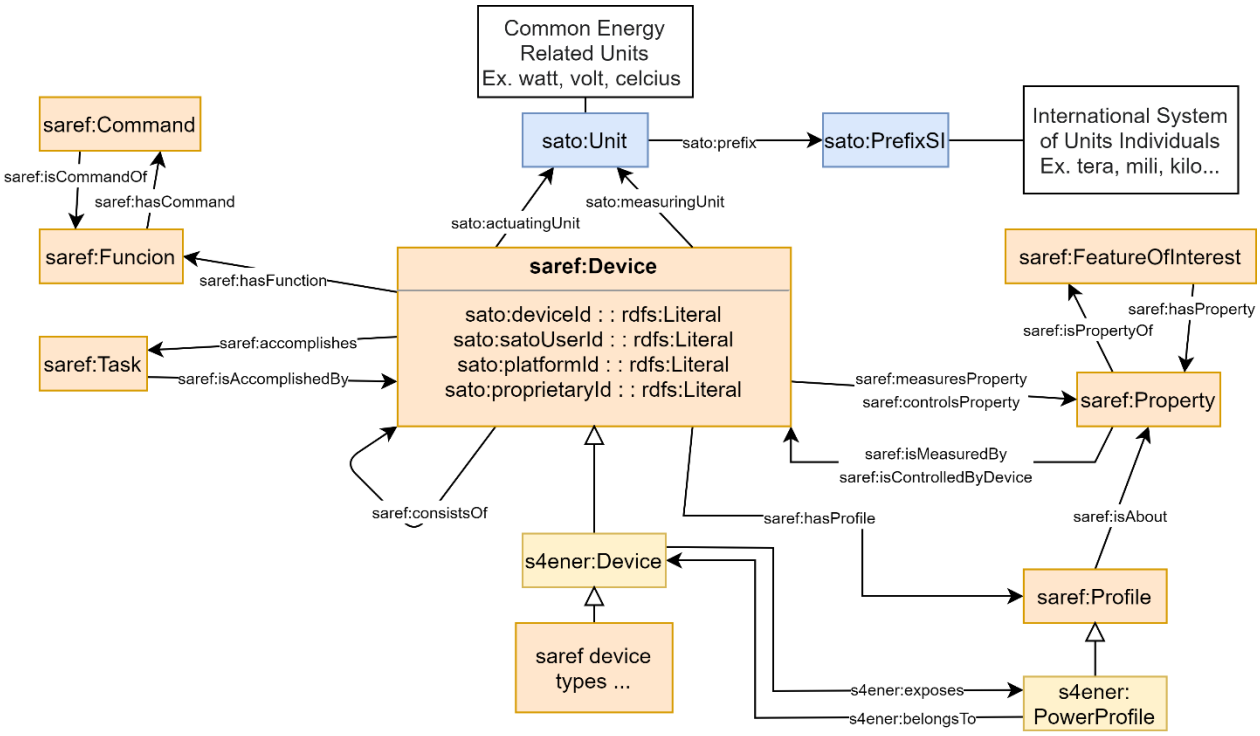

Figure 7 - SATO device ontology.

This ontology is based on the SAREF4ENER ontology [9], which standardizes the device representation independently of the underlying platform. This standardization covers, for instance, the description of the device's properties, tasks, and functions, allowing for a complete representation of the devices given the SATO project domain.

Figure 8 provides an example of the representation of a smart plug according to the Devices Semantics (SATO ontology). This representation includes the capabilities of the device, which indicates that the present smart plug has an ON/OFF function and a measurement capability. Both functions are associated with the power property of the device, meaning that the smart plug can measure the power consumption and actuate over the power state of the plug. This example shows the potential of the ontology in hiding the proprietary specification of the device, providing an integrated representation of devices' features.

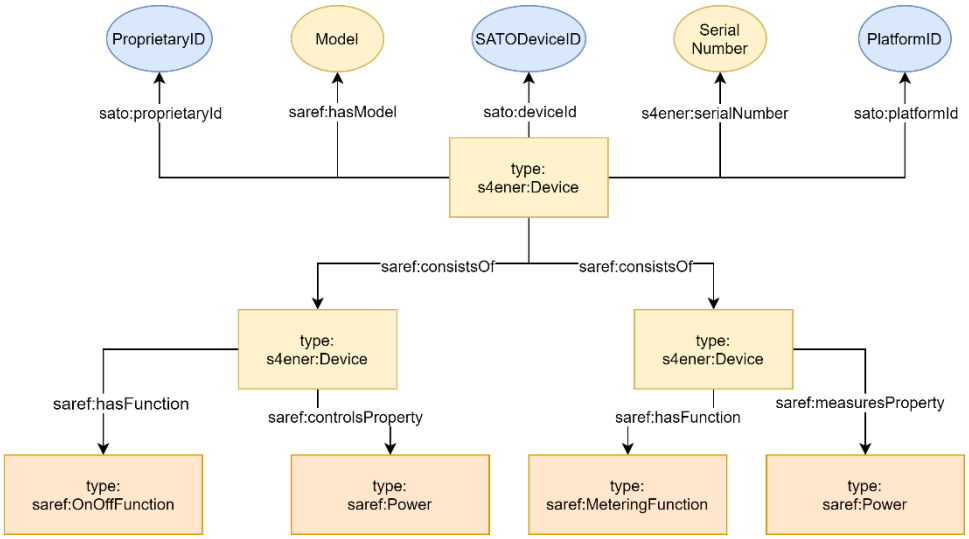

Figure 8 – Example of a smart plug represented in the SATO ontology.

The Device Semantics component has a database of triples where the information about the registration of a new device is stored. <http://example/devices/satoDeviceId-1> rdf:type s4ener:Device.

```
<http://example/devices/satoDeviceId> saref:consistsOf <http://example/devices/satoDeviceId-1>.

<http://example/devices/satoDeviceId-1/properties/1> rdf:type saref:Power.

<http://example/devices/satoDeviceId-2> rdf:type s4ener:Device.

<http://example/devices/satoDeviceId> saref:consistsOf <http://example/devices/satoDeviceId-2>.

<http://example/devices/satoDeviceId-2/properties/2> rdf:type saref:Power.

<http://example/devices/satoDeviceId-1/functions/0> rdf:type saref:MeteringFunction.

<http://example/devices/satoDeviceId-1>  saref:hasFunction  <http://example/devices/satoDeviceId-
1/functions/0>.

<http://example/devices/satoDeviceId-1>                                    saref:measuresProperty
<http://example/devices/satoDeviceId-1/properties/1>.

<http://example/devices/satoDeviceId-2/functions/0> rdf:type saref:OnOffFunction.

<http://example/devices/satoDeviceId-2>  saref:hasFunction  <http://example/devices/satoDeviceId-
2/functions/0>.

<http://example/devices/satoDeviceId-2>                                    saref:controlsProperty
<http://example/devices/satoDeviceId-2/properties/2>.

}
```

Figure 9 shows a query example of the smart plug registration.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX owl: <http://www.w3.org/2002/07/owl#>

PREFIX sato: <http://www.semanticweb.org/satoOnt#>

PREFIX s4ener: <https://saref.etsi.org/saref4ener/>

PREFIX saref: <https://saref.etsi.org/core/>


INSERT DATA {

<http://example/devices/satoDeviceId> rdf:type s4ener:Device;
```

```
        sato:deviceId "satoDeviceId";
        sato:proprietaryId "proprietaryId";
        sato:platformId "platformId";
        sato:satoUserId "satoUserId".

<http://example/devices/satoDeviceId-1> rdf:type s4ener:Device.
<http://example/devices/satoDeviceId> saref:consistsOf <http://example/devices/satoDeviceId-1>.
<http://example/devices/satoDeviceId-1/properties/1> rdf:type saref:Power.
<http://example/devices/satoDeviceId-2> rdf:type s4ener:Device.
<http://example/devices/satoDeviceId> saref:consistsOf <http://example/devices/satoDeviceId-2>.
<http://example/devices/satoDeviceId-2/properties/2> rdf:type saref:Power.
<http://example/devices/satoDeviceId-1/functions/0> rdf:type saref:MeteringFunction.
<http://example/devices/satoDeviceId-1>  saref:hasFunction  <http://example/devices/satoDeviceId-
1/functions/0>.
<http://example/devices/satoDeviceId-1>                                    saref:measuresProperty
<http://example/devices/satoDeviceId-1/properties/1>.
<http://example/devices/satoDeviceId-2/functions/0> rdf:type saref:OnOffFunction.
<http://example/devices/satoDeviceId-2>  saref:hasFunction  <http://example/devices/satoDeviceId-
2/functions/0>.
<http://example/devices/satoDeviceId-2>                                    saref:controlsProperty
<http://example/devices/satoDeviceId-2/properties/2>.
}
```

Figure 9 – Example of a smart plug registration into the SATO ontology.

From this example, we can see that all metadata concerning the smart plug is stores in the SATO platform. Afterwards, any component from the SATO platform can request information from the device semantics component to understand what features that platform supports.

Figure 10 represents an additional example of how to retrieve information from the Devices Semantics. This example lists the SATO IDs of the devices that have the power ON/OFF capability.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX sato: <http://www.semanticweb.org/satoOnt#>
PREFIX s4ener: <https://saref.etsi.org/saref4ener/>
PREFIX saref: <https://saref.etsi.org/core/>

SELECT ?deviceId
WHERE
{
  ?device rdf:type s4ener:Device;
          saref:consistsOf ?subDevice;
           sato:deviceId ?deviceId.
  ?subDevice saref:measuresProperty ?property;
```

```
            saref:hasFunction ?function.
  ?property rdf:type saref:Power.
  ?function rdf:type saref:OnOffFunction.
}
```

Figure 10 – Example of request information from the SATO ontology.

These two examples demonstrate how the interaction with the Device Semantics works. The same logic is used for the registration or retrieving data from these devices, platforms, or soft sensors.

When working with data from the EPREL database as a soft sensor in the SATO platform, several steps are executed for registering a device through the Device Services, namely:

1. When a device is registered, it triggers the connector that first checks, by comparing the registration number, if the device model data is already stored in the Metrics and Metadata database, and if not, its fetches the EPREL data. Otherwise, no request is issued.
2. The EPREL data is used to populate the ontology on the Device Semantics component. A device identifier in the SATO platform (`SATO ID`), which is assigned by the Data Catalog (CDM), is used to correlate the device with the semantic information.
3. The device `id` is sent through the system and used to insert the EPREL data on the Structured Storage component.

Currently, the insertion of the EPREL data for a device in the SATO platform will happen when registering the device in the SATO platform. This registration must contain the device product group code and the registration number, obtained from the QR code to be used in the API call *`Get product group's model by registration number`*.

The connector is used to make a fetch request for the EPREL model data through the EPREL API and then to store the data on the metrics and metadata database.

7. Common data model

The integration of heterogeneous commercial platforms and smart devices results in combining multiple diverse data formats for each one of them, hampering the comprehension and utilization of their data by the SATO components. To solve this problem, a unique data format was needed so that the other components only need to deal with a Common Data Model (CDM).

Since there are multiple data formats and the SATO project directly relates to the energy domain, the proposed architecture defines a CMD based on the EEBUS standard. However, only the SPINE [10] [11] part is used since it is oriented to data. Figure 11 shows an example of the representation of the CDM device description of a Smart Plug that is being used in the SATO platform.

As the example demonstrates, the physical device contains a device address (the SATO device ID) and a device type (which can be extended to cover more devices). However, this description is not sufficient to describe the device since most of the devices have physical or logical subparts that have similar capabilities. To separate the capabilities by its context, the device model in the CDM is composed of entities that will represent these subparts. This way, a Measurement feature can be interpreted in a different way if it is associated with a `TemperatureSensor` entity type or a `SubMeterElectricityType`.

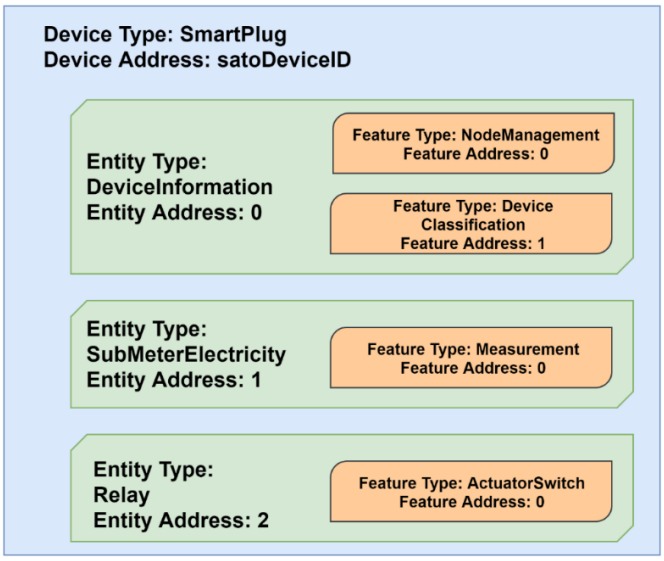

Figure 11 – CDM device description.

To conclude the standardization of the CDM, there are the feature types that are the most rigid definitions from the SPINE specification. These features can be seen as an entity function, where SPINE defines its data format. This way, events referencing the same feature type will have the same format. For the CDM to be adopted, each of the underlying data formats should be mapped to the CDM in a way that all components of the SATO platform can consume them easily.

Considering this specification, the CDM should have a structured message format capable of being generic for all its use cases. Since the EEBus SPINE specification is being used, the used structure is already defined covering the SATO platform needs. This structure is composed of a message header that identifies the device by its `SATODeviceID`, the entity to which the device applies to and the corresponding feature, the timestamp of the message, the classifier of the message that defines the type of message it is, and the SPINE specification version being used. To complement the header, there is the message payload that contains the actual data of the message, for example the event value. This payload contains the function element, which identifies the function in the message, and the function object itself which will contain the data according to the function specification in SPINE. This message structure is presented in Figure 12.

```
{
    "payload":{
        "cmd":{
            "function": functionId,
            functionId: functionObject
        }
    },
    "header":{
        "addressSource":{
            "feature": featureId,
            "device": satoDeviceId,
            "entity": entityId
        },
        "cmdClassifier":"notify",
        "specificationVersion": spineVersion,
        "timestamp": value
    }
}
```

Figure 12 – Example of data format generated by the CDM component.

When considering external data sources for the SATO platform, such has weather or energy classification, there are multiple data sources for the same context, where the same problem will arise regarding the heterogeneous data formats. In this sense, these data sources should also be standardized to facilitate their integration into the platform. To achieve it either the injected data has a soft sensor where most of the CDM design supports since it is focused on the concept of a device, or these data sources are not possible to be represented as soft sensors and then there should be a specific CDM for that type of data.

8. Data pre-processing

A second part or the SATO middleware block receives data in the CDM format, pre-processes it, and delivers it to the self-assessment and self-optimization layers of the SATO platform. The next subsections describe the main components responsible for this task.

8.1. Second-level streaming

Similar to the streaming instances presented in Section 5, the second-level streaming bridges the ingested data standardized by the Data Catalog component and the SAF. Before reaching the SAF and the Refined Storage component, data passes through a series of quality control and enhancement steps.

Some differences are noticeable from the data ingestion streaming instances (Section 5). First, the second-level streaming instance deals with both control and data events. We adopted this strategy because control events take effect when the Devices Service (Section 6) consumes and processes them. Control events sent to this streaming instance serve for analysis/statistical purposes or to inform optimization services about changes in the configurations of building devices. Second, events may be consumed, assessed, enhanced, and re-inserted into the second-level stream. This approach allows events to have their quality assessed and to be enhanced with additional meta-data, before being stored in the Refined Storage or sent to the SAF. Additionally, this strategy enables different events to take different flows, passing only through the components they are required to. For instance, an event that is not configured for data quality assessment, can flow directly to the data enhancement before being stored. These decisions will be taken according to performance, data utility, and quality level requirements for each type of data.

8.2. Data quality

The data quality component is meant to provide dependability monitoring capabilities to the SATO platform. Specifically, the component aims to improve the reliability and resilience/dependability properties of the platform, by applying a series of assessments on the data streams that reach the component. It is designed as an independent framework that can compensate data quality degradation, e.g., due to sensor faults causing drift or outliers, time-variability of processes, or communication failures.

The generalized framework is illustrated in Figure 13. From an operational point of view, it receives sensor measurements originated in buildings and provides corrected measurements and an associated data quality assessment or index. Both the original measurement and any corrections, are stored in the SATO platform.

For that, the framework considers a series of sensor failure modes, application domain knowledge, and generalized machine learning methodologies, that can be used for decision making or model design. The operation is based on data fusion techniques using machine learning that model each sensor behavior according to a set of sensors past measurements. Considering that there must be an history of measurements before the application of the framework procedures, there is a preliminary step consisting of the creation of such models. For fault-detection purposes, each sensor must be represented by at least two models exploring temporal, spatial and value correlations between target sensor past measurements or a combination of sensors existent in the building.

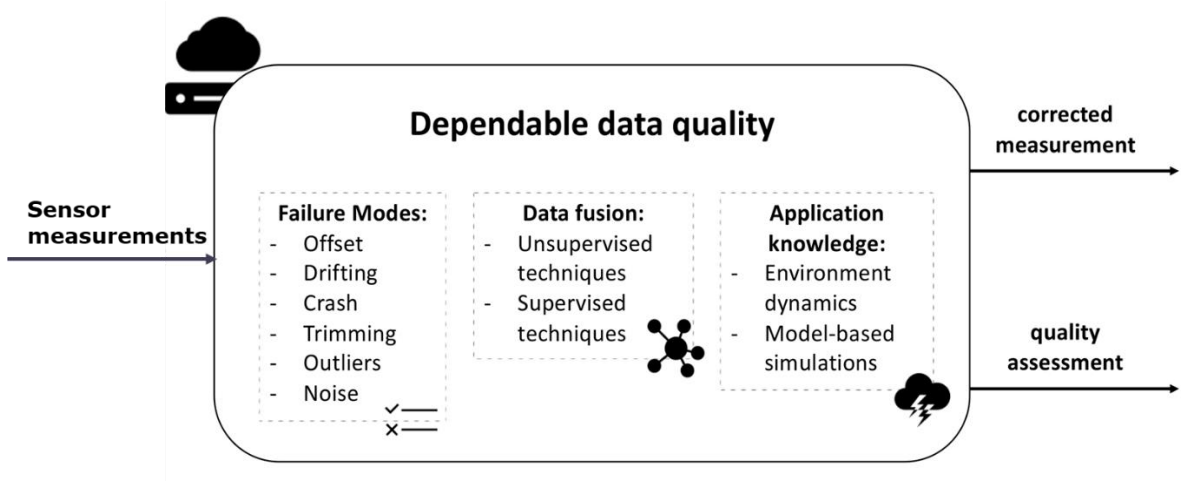

Figure 13 - SATO data dependability framework.

The basic architecture of the SATO data quality assessment is shown in Figure 14. It is composed of four main modules: Failure Detection (FD), Prediction (P), Quality Evaluation (QE), and Measurement reassessment (MR).

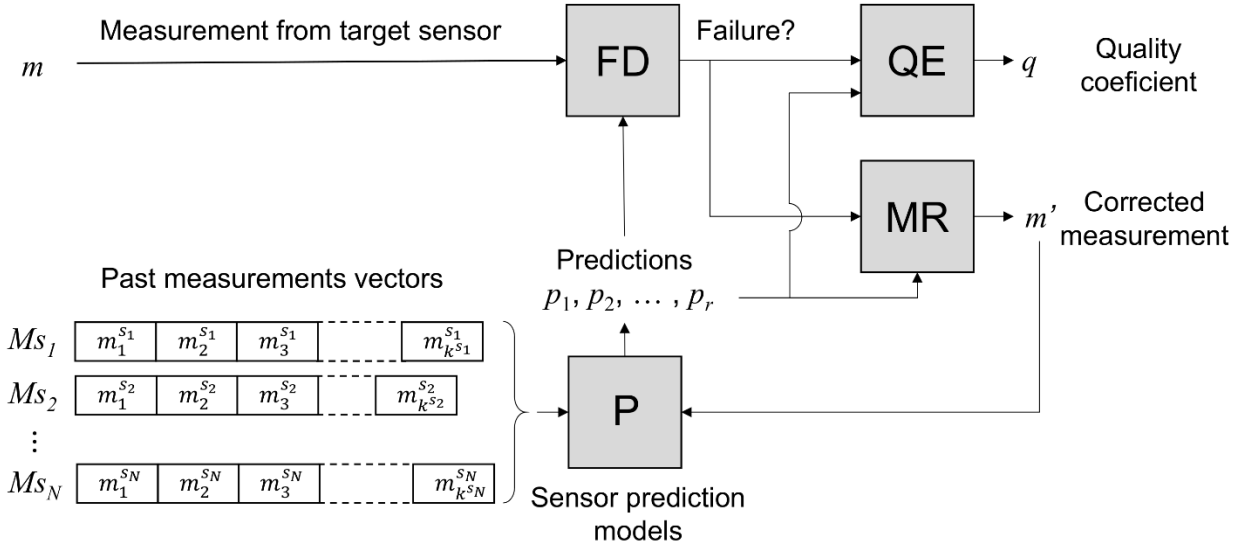

Figure 14 - Data quality assessment architecture.

The truthfulness or quality of a measurement can only be assessed by knowing the involved impacting conditions, what is seldomly feasible. Alternatively, to substantiate the truthfulness of a measurement, an attainable method is to employ several prediction methods to estimate the measurement. This is accomplished in the P module.

In the FD module, the objective is to detect and identify failure behaviors affecting measurements of the target sensors. This involves the execution of procedures not only to characterize abnormal measurements but also to distinguish sensor faults from environment-related events. The predictions obtained in the previous module are of great importance in this step.

The outcome of the previous components enables the quantification of the confidence level on the new received measurement. The QE module output is a quality coefficient associated to the measurement. When facing a failure behavior, this value is set to 0.

If a measurement is faulty, it is important for future predictions and for further systems to have access to an estimate of the expected measurement, even if it is not fully reliable (confidence level below standard). The MR module enables the mitigation of the effects of the failure behavior by producing an estimation of measurement as if it had not been affected by any fault.

8.3. Data enhancement

Considering that most IoT platforms and smart systems only send bare minimum data needed to represent a device event, this information will not be very rich when it becomes standardized by the SATO Data Catalog component. So, to achieve a better assessment of the data received in the platform, the Data Enhancement component should interpret the data consumed by the Streaming component and enrich it in any way possible using the Device Semantics information stored since the device registration.

8.4. Refined storage

The Refined Storage component is considered the trustworthy persistent storage of all events coming from the devices and buildings or the external data sources. Data stored in this component will have passed through the Data Catalog component to be standardized using the Common Data Model (CDM) and, opportunistically, will have passed through the Data Quality and Data Enhancement components (when configured for doing so).

Data stored in the Refined Storage will serve as the input for the Self-Assessment Framework (SAF), which will be described in the next section. This data may contain standardized formats of raw measurements and actions that may contain privacy-sensitive data. For that reason, data availability, access control, and appropriate security premises are a must for this component since it will store the main portion of data to be processed.

9. Self-assessment framework

The SAF is the block of components that will perform the main processing tasks towards estimating the SRI of buildings and calculating other energy or energy-related building assessments that will support the components in the self-optimization block. At this stage, multiple computations will take place using the standardized data that is available at the Refined Storage. The following subsections briefly describe the components of the SAF. Deliverable 3.8 will provide more details about the SAF architecture.

9.1. Data structuring and structured storage

The first step towards providing the appropriate computing platform for the SAF encompasses preparing the data for the required data structures and databases. This step, called Data Structuring, consists of converting and organizing the standardized data available in the Refined Storage into formats and data structures appropriate for processing. The resulting structured data will then be sent to the Structured Storage component, becoming available to the Assessments component.

9.2. Location and occupation services

The objectives of the Location and Occupation services component is to implement an assessment that continuously estimates the number of occupants and their spatial distribution in buildings, and locates certain specific sensors deployed in buildings. By doing so at regular time intervals it allows building managers to better understand occupancy flows and energy management services to improve the management of building energy resources.

For this purpose, Wi-Fi and Bluetooth connectivity signals are collected in buildings, either from access points or by means of dedicated scanners. These signals and corresponding signal strengths will be non-

intrusively collected from occupants' devices, such as computers, smartphones, and other wearables. After being de-identified and anonymized, the connectivity information is sent to the SATO platform data ingestion streaming component, as from any other sensor. It must be mentioned that at that point it is no longer possible to trace-back the information to the specific device that generated it.

When the Location and Occupation component receives the data from the Structured Storage component, it will merge the data and use a machine learning algorithm to estimate the location of each signal. These results are fed back to the platform data streaming component to reach the Structured Storage component, becoming available to the Assessments component where the locations are used to assess occupancy and spatial distribution. The location of specific sensors connectivity signals are directly used to pin-point the location of EBCs of special interest, so they can be dynamically displayed in BIM interfaces

9.3. Building and assessment semantics

The Building Semantics component, similarly to the Device Semantics, structures buildings data according to an established ontology. The main objective is to define concepts related to buildings and the respective information relations. It may include multiple domains, e.g., the building topology, where it should describe the building structure to allow location in the topology. Another domain to cover is the building types, allowing to distinguish assessments and optimization services accordingly.

The Assessment Semantics is another ontology component that will define relevant concepts of energy assessment in buildings. It will enable the SATO platform to describe all assessments in a standardized way in relation to devices and buildings, allowing easier definition of self-assessment and self-optimization services. This ontology is expected to be an original proposal of the SATO Project since no previous complete ontological solution focusing on buildings energy assessment was found in literature.

Regarding implementation, the interaction between device semantics, building semantics, and energy assessment semantics, there are two possibilities. Either the ontologies are joined, having only a logical separation in the architecture representation, or the connection between the ontologies is done through IDs, where, for instance, a device has a zone ID or a building zone has multiple device IDs.

With the ontologies in place, the platform services can be extended. For instance, the Device Services component can have a richer set of services, such as powering off the devices in a specific zone of a building or increase the heating set-point in all devices of a building. To conclude, the semantic components enable an easier definition of an extended set of services, providing additional services than those the individual integrated devices and platforms would allow.

9.4. Self-assessments

Multiple building energy assessments will be implemented by processing the data available in the various storage components of the platform. These assessments focus mostly on estimating the SRI of buildings and providing intermediate energy performance results for the subsequent interactive applications and optimized energy management services. The results from this component, together with the self-optimization services, represent a considerable portion of the added value that the SATO platform will deliver for all stakeholders.

Although this component is conceptually represented as a single element in Figure 1, it complexity extrapolates the level of description from this Deliverable. Ongoing tasks from Work Package WP3 (Development of SATO self-assessment framework) are discussing which and how the self-assessments will be implemented in practice. Moreover, Deliverable D3.1 will contain a detailed description of the several self-assessments the SATO platform will provide, the system identification toolbox that will support some of the assessments, and the specific computing requirements they demand.

9.5. Results storage

The Results Storage is a component that will collect the outcomes from the self-assessments and will provide them to the upper layers of the architecture, namely self-optimization services and actors. These results will include (but are not limited to) the estimation of the SRI of buildings, the multiple variables that compose the SRI, EBC energy performance assessments, statistics about the building status and occupancy, among others. Another important example of data that will be stored in this component are the BIM files, which are going to be generated on-demand when users want to visualize information from the monitored buildings using the BIM visualization tools.

Although the Result Storage component is apparently a simple, single component in Figure 1, it may contain several storage configurations, tools, and drivers. These different storage instances can provide the appropriate service levels and security protection according to specific data types and data utilization.

10. Self-optimization services

The Self-optimization Services block of components will be used for taking manual or automatic energy management actions to the building devices, and to facilitate interaction with the relevant Actors. The next subsections briefly explain the purpose of each component.

10.1. SATO apps

This component abstracts the set of web-based or mobile applications that will be developed in WP5 of the SATO project. They will provide the desired level of interaction with the self-optimization services to the actors involved in the building's energy management. These actors will be able to provide their preferences and configurations that, ultimately, will be considered when the self-optimization services decide upon the most appropriate controls for buildings and their devices.

10.2. BIM interface

The BIM is a specific visualization interface that the SATO platform will provide integrated with the remaining data flow. This component presents three-dimensional multi-layer information about the buildings (and their devices) through a BIM visualization tool.

An up-to-date version of the BIM model representing the whole building can be prepared periodically (e.g., every 15 minutes) and stored in the Result Storage component. Alternatively, it can also be produced on-demand. Having access to such an enriched interface facilitates the job of building managers since they can evaluate the current state and behavior of the building before taking decisions, as well as overview the impact of recent actuations and decisions on the building and its devices.

10.3. Self-assessment services

The self-assessment services component provides enhanced dashboards and aggregated metrics based on the assessment results in the SAF. It may be viewed as a non-BIM interaction and visualization part of the assessments component in the SAF. These services to be developed in WP3, will enable building managers and other actors to easily obtain contiuously updated reports on all metrics evaluated by the SATO platform.

10.4. Self-optimization services

The self-optimization services component aggregates the set of energy management services that will be developed in WP4 of the SATO project. Each of these services will be implemented as a program or set of programs that will use data stored in the Results Storage (Section 9.5) component as input to determine the control actions that will optimize energy resources in buildings, thereby demonstrating how the SAF can contribute for increased energy flexibility, efficiency, and user satisfaction.

The computed control actions will be sent to the Devices Services (Section 6) component that will translate the desired actions into device-specific formatted commands that will finally be sent to the building devices. These decisions might be based on optimization procedures, rule-based methodologies, or on experience-based decisions from the responsible actors (e.g., building managers).

In the latter case (i.e., user-based decisions), the responsible actors will interactively make decisions based on the applications, dashboards, and aggregating services mentioned in the present section. These components will also consume data from the Results Storage and the control actions will also be sent to the Devices Services.

In both scenarios, the self-assessment and optimization (SA&O) services and applications will only have limited access to data through the appropriate interfaces and contexts. For instance, it differs from the broad access that the Self-assessment framework will have to the data stored in the lower-level storage components.

Decisions at this stage can be proactive or reactive. In the former, the optimized energy management service or the actor make the decision periodically based on the current state of the buildings and predefined goals. They do not wait for a triggering event to indicate the need for adapting. In the latter, the optimized energy management service or the actor take the decision reactively according to some triggering event that demands adapting the building to correct some property.

Independently on the case, the Devices Services receives actuations and forwards them to the appropriate building and devices through direct communication with the devices or through the respective channels in the streaming component for control events. Once the actuation takes place, the devices (and consequently, the buildings) will adapt accordingly. The actuations sent to the buildings are also forwarded to the Transient Storage and the Data Catalog component, for the platform to store information about the previously taken decisions.

11. Conclusion

This deliverable presents the overall conceptual architecture of the SATO platform. It organizes the platform into six building blocks: external data sources, buildings, SATO middleware, self-assessment framework, self-optimization services, and actors. These blocks are presented in Figure 1, identifying and describing also their internal conceptual components and data flows.

Many building blocks, components, and data flows have already been defined by Task T1.3 (Requirements and system architecture for the SATO platform and for it to support SRI calculation), which concludes with this deliverable. The definitions presented in this document initially focused mostly on the data obtention from buildings and other external data sources to their respective provision and storage into the SATO Middleware. Additionally, the description includes all other components foreseen in the SATO architecture and initiated the discussion of their internal architecture. The focus was to present multiple conceptual components that will be implemented in SATO WPs and described in future deliverables.

Bibliography

[1] O. Miralles, "European Product Registry for Energy Labelling (EPREL) Compliance Site Description," 2018. [Online]. Available:

https://webgate.ec.europa.eu/fpfis/wikis/display/EPREL/EPREL+Compliance+Site+Description. [Accessed 24 06 2021].

[2] EPREL, "REGULATION (EU) 2017/1369 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL," 4 July 2017. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv%3AOJ.L_.2017.198.01.0001.01.ENG.

[3] "About the energy label and ecodesign," [Online]. Available: https://ec.europa.eu/info/energy-climate-change-environment/standards-tools-and-labels/products-labelling-rules-and-requirements/energy-label-and-ecodesign/about_en. [Accessed 24 06 2021].

[4] "EPREL Product database | European Commission," [Online]. Available: https://ec.europa.eu/info/energy-climate-change-environment/standards-tools-and-labels/products-labelling-rules-and-requirements/energy-label-and-ecodesign/product-database_en.

[5] "Regulation on energy labelling for electronic displays (EU) 2019/2013," 11 March 2019. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2019.315.01.0001.01.ENG&toc=OJ:L:2019:315:TOC.

[6] O. MATHIEU, J.-L. JARACZEWSK, J. T. DE NUNES and D. A. ACCOGLI, "EPREL Exchange Model Documentation," 9 6 2021.

[7] "EPREL API," [Online]. Available: https://webgate.ec.europa.eu/fpfis/wikis/display/EPREL/EPREL+Public+site+-+API. [Accessed 23 06 2021].

[8] W3C, "Resource Description Framework (RDF) Model and Syntax Specification," [Online]. Available: https://www.w3.org/TR/PR-rdf-syntax/.

[9] L. Daniele, "SAREF4ENER: an extension of SAREF for the energy domain created in collaboration with Energy@Home and EEBus associations," 04 June 2020. [Online]. Available: https://saref.etsi.org/saref4ener/v1.1.2/. [Accessed 14 July 2021].

[10] EEBus Initiative e.V., "EEBus SPINE Technical Specification Protocol Specification," Cologne, 2018.

[11] EEBus Initiative e.V., "EEBus SPINE Technical Specification Resource Specification," Cologne, 2018.

[12] I. Esnaola, F. J. Diez, N. Tomasevic and M. Batic, "RESPOND D4.1 Semantic Information Model," 2019.

[13] The Apache Software Foundation, “Apache Jena,” [Online]. Available: https://jena.apache.org/. [Accessed 30 March 2021].

[14] D. Ö. Thomas Märzinger, "MDPI," 22 May 2019. [Online]. Available: https://www.mdpi.com/1996-1073/12/10/1955/htm.

[15] R. P. G. P. A. G. Ilaria Vigna, "MDPI," 1 June 2020. [Online]. Available: https://www.mdpi.com/1996-1073/13/11/2796/htm.

[16] D. Ö. Thomas Märzinger, "MDPI," 7 June 2020. [Online]. Available: https://www.mdpi.com/1996-1073/13/13/3507.

[17] "SAREF," ETSI, 29 5 2020. [Online]. Available: https://saref.etsi.org/core/v3.1.1/.

[18] schwackenberg, "EEBus SPINE Technical Report," Cologne,, 2018.

[19] Microsoft, "Docs.microsoft," Microsoft, 13 1 2021. [Online]. Available: https://docs.microsoft.com/en-us/dotnet/architecture/microservices/multi-container-microservice-net-applications/integration-event-based-microservice-communications.

[20] M. S.-S. R. P. Dmitry Namiot, "Researchgate," 22 Oct 2018. [Online]. Available: https://www.researchgate.net/publication/327936728_On_Data_Stream_Processing_in_IoT_Applications_18th_International_Conference_NEW2AN_2018_and_11th_Conference_ruSMART_2018_St_Petersburg_Russia_August_27-29_2018_Proceedings.

[21] J. Yates, "Stream Processing with IoT Data: Challenges, Best Practices, and Techniques," 4 June 2020. [Online]. Available: https://www.confluent.io/blog/stream-processing-iot-data-best-practices-and-techniques/.

[22] J. Spacey, "Simplicable," 27 12 2016. [Online]. Available: https://simplicable.com/new/separation-of-concerns.

[23] C. Schults, "What Is Infrastructure as Code? How It Works, Best Practices, Tutorials," 5 9 2019. [Online]. Available: https://stackify.com/what-is-infrastructure-as-code-how-it-works-best-practices-tutorials/.

[24] Ionos, "Infrastructure as Code (IaC): using code for IT infrastructure management," 14 10 2019. [Online]. Available: https://www.ionos.com/digitalguide/server/know-how/infrastructure-as-code/.

[25] A. Patel, "Configuration Management and Continuous Deployment," 12 4 2018. [Online]. Available: https://medium.com/formcept/configuration-management-and-continuous-deployment-cd0892dce998.

[26] J. Harris, "The growing importance of big data quality," 21 11 2016. [Online]. Available: https://blogs.sas.com/content/datamanagement/2016/11/21/growing-import-big-data-quality/.

[27] Stitch, "What is a Data Lake? Examples & Solutions," [Online]. Available: https://www.stitchdata.com/resources/what-is-data-lake/.

[28] European Parliament, Council of the European Union, "Directive (EU) 2018/844," 30 May 2018. [Online]. Available: https://eur-lex.europa.eu/eli/dir/2018/844/oj. [Accessed 2018].

[29] OWASP Foundation, "OWASP/DevGuide," [Online]. Available: https://github.com/OWASP/DevGuide.

[30] Y. M. P. V. T. S. B. V. G. O. S. E. P. W. :. K. B. J. A. A. H. M. O. J. G. :. M. U. J. S. Stijn Verbeke, "SUPPORT FOR SETTING UP A SMART READINESS INDICATOR FOR BUILDINGS AND RELATED IMPACT ASSESSMENT," European Commission, 2018.

[31] Fraunhofer, "OGEMA - Open Gateway Energy Management Alliance," [Online]. Available: https://www.ogema.org/.

[32] FIWARE Foundation, "FIWARE - The Open Source platform for our smart digital future," 2021. [Online]. Available: https://www.fiware.org/.

[33] "Apache Kafka," [Online]. Available: https://kafka.apache.org/.

[34] "Eclipse Mosquitto," [Online]. Available: https://mosquitto.org/.

[35] "RabbitMQ," [Online]. Available: https://www.rabbitmq.com.

[36] "Docker," [Online]. Available: https://www.docker.com/.

[37] EEBus Initiative e.V., "EEBus SPINE Technical Report Introduction," Cologne, 2018.