D4.2 Enhanced, user-friendly EMS for residential/building flexibility Discovery and delivery (v2)

31 October 2025

OPENTUNITYproject.eu

Deliverable details

Title	WP	Version
Enhanced, user-friendly EMS for residential/building flexibility Discovery and delivery	4	2.0

Contractual delivery date	Actual delivery date	Delivery type*	Dissemination**
M34 (October 2025)	M34 (October 2025)	DEM	PU

*Delivery type: R: Document, report; DEM: Demonstrator, pilot, prototype; DEC: Websites, patent fillings, videos, etc; OTHER; ETHICS: Ethics requirement; ORDP: Open Research Data Pilot.

Dissemination Level: **PU: Public; **CO**: Confidential, only for members of the consortium (including the Commission Services)

Author(s)	Organization
Panos Andriopoulos	HYPERTECH
Panagiotis Tsatsoulis	HYPERTECH
Giorgos Giannakis	HYPERTECH
Kosmas Petridis	HYPERTECH
Panos Sabatakos	HYPERTECH

Version	Date	Person	Action	Status***
1.1	18.07.2025	Panos Sabatakos	Section 4 update (BFMS)	Draft
1.2	12.09.2025	Kosmas Petridis, Pablo Bort, Tomaž Buh, Federico Giani	Section 2 update (HEMS/BEMS)	Draft
1.3	17.10.2025	Pablo Bort	Section 4 update (NILM)	Draft
1.4	22.10.2025	Panos Sabatakos	Section 4 update (BFMS)	Draft
1.5	30.10.2025	Sara Vieira, Oluwakorede Adenuga	Peer review	Final

2.0	31.10.2025	Tsatsoulis Panagiotis	Version ready to submit	Submitted

***Status: Draft, Final, Approved, Submitted (to European Commission).

Authors (organization)
Pablo Bort (ETRA I+D); Álvaro Nofuentes (ETRA I+D); Tomaž Buh (REDUXI); Federico Giani (AEM)

Keywords

Demand Response (DR), Load Flexibility, Energy Management, Grid Interaction, Energy Optimization, Forecasting Algorithms, Load Prediction, Flexibility Modeling, Real-time Data Analytics, Smart Buildings, IoT Sensors, Load Monitoring

Executive Summary

This Deliverable presents the development of the Building Flexibility Management System (BFMS) and the Non-Intrusive Load Monitoring (NILM) components of the OPENTUNITY project. These two components concern the interaction between Home Energy Management Systems (HEMS), Building Energy Management Systems (BEMS) and various flexibility market stakeholders, leading to disaggregation of energy loads and forecasting of baseline and flexibility power consumption.

The deliverable starts with a description of the HEMS/BEMS that are available at the four pilot sites of the project. These systems are necessary tools for the calculation and dispatch of flexibility, as they are responsible for the seamless communication of data from the pilots, as well as the dispatch of control signal based on flexibility forecast. The main functionalities of the available HEMS/BEMS are presented, focusing on hardware/software setup, performed data collection, supporting IoT devices and interfaces (where available).

BFMS collects energy usage data, sensor information and occupant preferences to forecast baseline consumption and available flexibility. By analyzing this data and using machine learning algorithms, BFMS determines how much energy can be adjusted and when, enabling buildings to participate in Demand Response (DR) programs. It then dispatches control signals to connected devices, ensuring energy is used efficiently while respecting user constraints. The DR Initialization Service is a novel module of BFMS, used to collect occupant feedback on available assets and comfort preferences. It ensures a user-friendly experience, allowing consumers to participate in DR programs without compromising comfort or altering their behaviors.

The NILM component provides meaningful energy related insights by analyzing total power consumption and disaggregating it into individual appliance usage using variables like active power, reactive power, current and voltage. Unsupervised and semi-supervised methods like k-means clustering have been explored, offering a trade-off between reduced accuracy and the benefit of not requiring intrusive monitoring. In this way, real-time appliance power consumption can be predicted, creating generalized models that can be applied to any household with metering data.

By generating accurate consumption curves for flexible assets, NILM supports demand response programs, enabling more effective load management and energy market participation. Together, BFMS and NILM create a powerful tool for the monitoring and optimization of energy consumption, enabling greater engagement in demand response initiatives and flexibility calculation and dispatch. Actual end-to-end test cases were performed to illustrate the use of the developed tools in the framework of the OPENTUNITY project and its associated pilot sites.

Copyright statement

INDEX

List of figures

List of tables

1. INTRODUCTION

1.1 Purpose of the document

The purpose of the deliverable is to present the work carried out under Task 4.1 "HEMS and BEMS flexibility (including EV and storage) and DR optimization", Task 4.2 "Initial settings algorithm" and Task 4.3 "NILM and behavior analytics" of the OPENTUNITY project.

Task 4.1 focuses on extending and adapting Home Energy Management Systems (HEMS) and Building Energy Management Systems (BEMS) to fulfill the requirements of the project (e.g., integration of new interfacing protocols, adjustments for pilot site assets, etc.). In addition, the Building Flexibility Management System (BFMS) is also adjusted and extended based on the needs of the project. Task 4.2 complements the work of T4.1 by introducing a User Interface that allows users to provide their preferences and constraints, altogether enabling the dispatch of flexibility that considers the needs of the consumer. Lastly, Task 4.3 investigates current Non-Intrusive Load Monitoring (NILM) algorithms to select and adapt the best load identification techniques for the metering systems of the project. Following that, the selected NILM methodology is used for data acquisition, event detection, learning and load disaggregation.

The first version of the document was submitted in M26; this version is the updated and final version of the document providing the development of these three tasks by M34, which is the ending month of the three tasks.

1.2 Scope of the document

The deliverable details the activities undertaken to achieve the objectives of Tasks 4.1, 4.2 and 4.3. In a nutshell, this concerns the technical and modelling aspects for the forecast and dispatch of flexibility at the different pilot sites of the project, as well as the disaggregation of energy consumption to separate assets' consumption. Figure 1 provides a high-level representation of the OPENTUNITY energy flexibility services, including different components, tools and services that facilitate data collection from available assets and enable flexibility forecasting, bidding and trading. It is presented here to provide a clear picture of the way that the components developed under Tasks 4.1-4.3 are integrated into the OPENTUNITY ecosystem and of their interactions with the other tools of the project.

Figure 1 presents the different layers of this ecosystem, summarized as follows:

- the pilot sites with their respective assets,
- the HEMS/BEMS,
- the data exchange system,
- the flexibility forecasting tools, and
- the NODES market platform.

Starting with the pilot projects and data collection, the different sites of the project in Switzerland, Spain, Greece and Slovenia will be testing the integration of flexibility assets into the energy market. Each pilot site provides various flexibility assets, e.g., heating systems, air conditioners, domestic hot water heaters, etc. A distinct **HEMS/BEMS** manages these assets at each pilot, such as the ELIOT

(Spain), the Swiss pilot data platform (Switzerland), the Smart Box (Greece) and the Reduxi (Slovenia). These systems are presented in detail in Section 3 (OPENTUNITY HEMS/BEMS) of this deliverable. The HEMS/BEMS stream the necessary power consumption data from the pilots.

The Data Space serves as a central data exchange hub that ensures seamless communication between the various tools and services of the project. It is developed within WP3 of the project and is presented in the respective deliverables. Among other things, it ensures proper communication between the HEMS/BEMS and the tools of the project.

A suite of digital tools facilitates the identification, optimization and activation of flexibility resources. Based on data received by the pilot sites, BFMS calculates the baseline consumption of the available assets and forecasts their day-ahead flexibility provision. In addition, the **Demand Response (DR) Initialization Service** provides necessary information on the available assets and on the preferences of the users of these assets. These two tools are presented in Section 3 (Building Flexibility Management System) of this deliverable. The **NILM Load Identification Tool** detects individual electrical loads based on total energy consumption. It is presented in Section 4 (NILM and Behaviour Analytics) of this deliverable. The NILM tool is used in case no sub-metering is available at the consumers premises.

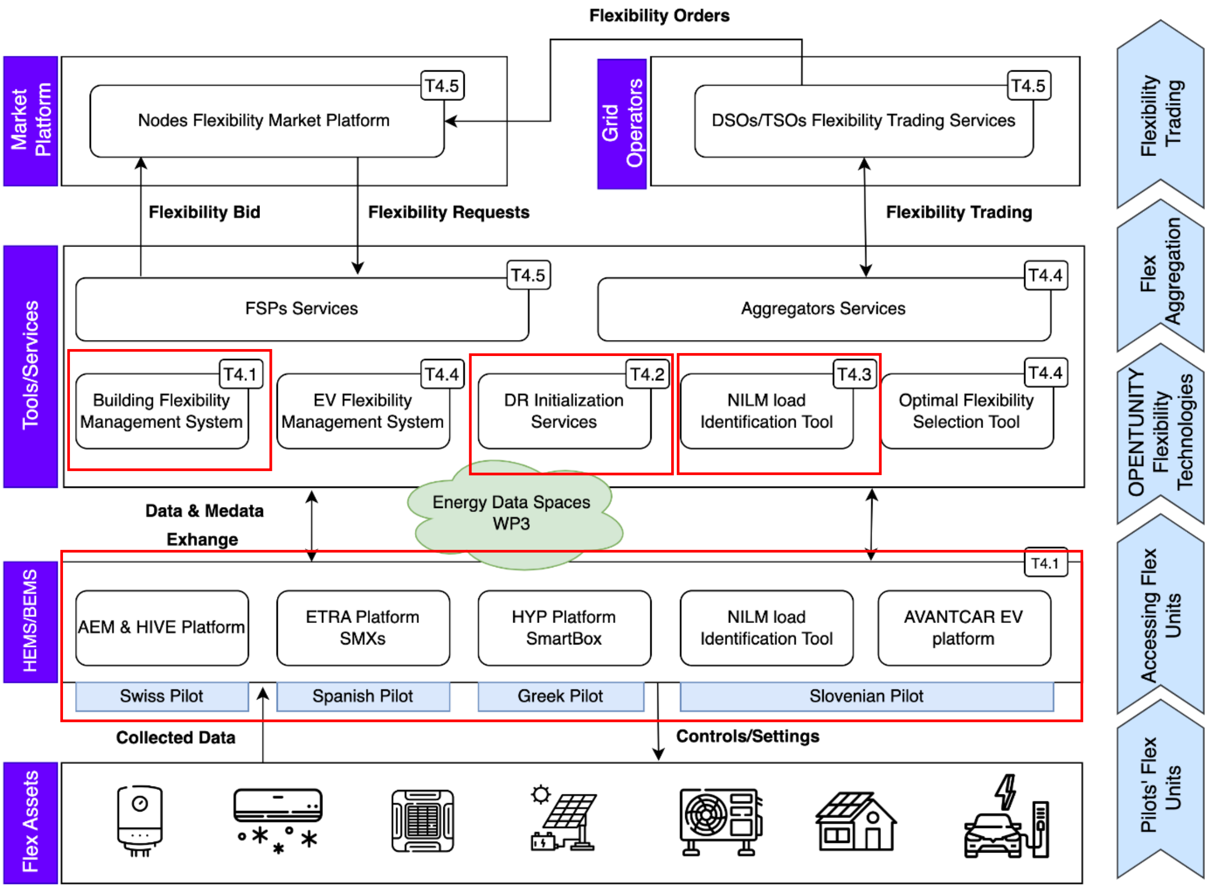


Figure 1: High-level representation of the OPENTUNITY services and tools

1.3 Structure of the document

The deliverable is structured as follows:

The first section, "Introduction", provides an overview of the deliverable, including its purpose, scope, structure and updates from the previous version. This section gives a brief overview of the deliverable, highlighting its objectives and relevance to the OPENTUNITY project in general and to the other tasks of WP4 in particular.

The second section, "OPENTUNITY HEMS/BEMS", focuses on the HEMS and BEMS that are used at the four pilots of the project. The section includes a description of the ELIOT, AEM R&D platform - Flexo, SMART BOX and REDUXI platforms, that are providing the necessary data for the development and use of the tools of WP4.

The third section, "Building Flexibility Management System", presents the methodologies and technical developments applied in Tasks 4.1 and 4.2. Here, several subsections provide details regarding the development process of BFMS, proposed methodologies, technology stack, input/output parameters and development status. The section also describes the assumptions made during the development of the tools, their restrictions and, lastly, presents examples illustrating their performance.

The fourth section, "NILM and Behaviour Analytics", presents work related to the NILM and behavioral analytics tool. Its structure is similar to the third section by including subsections on prototype development, methodologies, tools, interfaces and testing examples.

The fifth section, "Conclusions", presents the key findings and achievements presented in the document, while serving as a summary of the deliverable's contributions to the OPENTUNITY project.

1.4 Updates from the previous version

The second and last version of the deliverable entitled "Enhanced, user-friendly EMS for residential/building flexibility discovery and delivery" builds upon the first version, enhancing and expanding it to report the progress that has been made between M27 and M34 of the project. The final version incorporates all the sections of the previous version, updating or adding information to specific parts of the report.

Between M27 and M34, several activities were undertaken to advance the system design, integration and validation processes. These activities, which necessitated the present updates, are summarized below:

- **Refinement of updated HEMS/BEMS solutions (Section 2):** The Spanish pilot transitioned from the BESOS system to the updated ELIOT platform, including enhancements in interface design and the integration process. More information is provided for the Swiss pilot to include the latest information on asset monitoring and management.
- **Enhancement and extension of BFMS components and services (Section 3.3):** The Flexibility Forecasting Module, DR Initialization Service, Control Dispatching Service and Flexibility Service Provider were enhanced, implemented and tested to ensure their overall operation. These developments were complemented with new component and sequence diagrams illustrating the detailed functionality and interdependencies of the services.

- **Integration activities (Section 3.3.7):** Dedicated integration efforts were made to link the BFMS with the broader OPENTUNITY ecosystem, including the HEMS/BEMS deployed in Greece, Spain and Switzerland, the OPENTUNITY Data Space, and the NODES market platform.
- **Integration testing and data-driven validation (Section 3.7):** End-to-end validation activities were carried out using real pilot data, producing a series of graphs that demonstrate the interaction and performance of the BFMS components.
- **NILM tool enhancement (Section 4)**: A post-prediction analytics method was added to the NILM tool, refining disaggregation results by detecting and attributing power spikes to specific appliances. More figures and graphs were added to showcase the GUI and the general operation and performance of the tool.

In general, the second and last version of the deliverable describes the enhancement of the proof-of-concept version to a fully developed, operational and integrated system. It provides evidence of a working system capable of calculating, forecasting and dispatching building flexibility in a user-centric manner, backed by real data and detailed technical descriptions of the final components.

2. OPENTUNITY HEMS/BEMS

2.1 ELIOT

ELIOT[1] is a HEMS/BEMS from ETRA, its main functionalities are:

- Optimize the use of energy resources in the environment they operate,
- Optimize the time of use and general operation of the devices,
- Calculate and aggregate the flexibility of the assets and
- Provide predictive maintenance and diagnostics of the devices by means of data analytics

The software is continuously evolving and, up until now, it has been able to connect to different devices:

- Smart Meters,
- PV panels,
- Batteries,
- Electric Vehicle chargers and
- Smart Plugs

The connection to Smart Meters, PV panels and batteries has been done via Modbus TCP, periodically retrieving the status, measurements and control actions bi-directionally to the software platform through MQTT. Other PV panels data retrieval has been done using a Huawei platform API.

For EV chargers it depends on the already in use platform for charging strategies. In case the platform can use Open Charge Point Protocol (OCPP) standard, the charger is connected to a backend in the cloud deployed by ETRA and the information on transactions and charging parameters are automatically updated. If OCPP is not available, it has been interfaced using Modbus RTU.

Regarding smart plugs, they have been integrated using an instance of openHAB software running on-premises, which provides a RESTful API for retrieving the status and energy parameters. For smart plugs from the provider "Shelly", a direct connection to Shelly Cloud API has been used for data retrieval.

The Modbus and openHAB data collection have been performed using hardware from ETRA called SMX. This device has some CPU and RAM resources to run Docker containers where instances of openHAB and different SMX configurations and applications are hosted, such as the Graphical User Interface. The SMX can be configurated to connect to devices via Wi-Fi, Ethernet or Modbus.

The platform is hosted on a cloud server. It was built using Meteor, a JavaScript web development framework whose main advantage is its publish-subscribe pattern which feeds the clients with the

[1] The tool currently referred to as ELIOT is the updated version of the former BESOS tool in D4.1, with improvements made to its interface and integrations.

most updated information without requiring them to refresh the webpage. The GUI was developed also using React, a JavaScript library.

The main dashboard page can be seen in Figure 2, here the monitored buildings or households can be seen and KPIs regarding energy measurements are available.

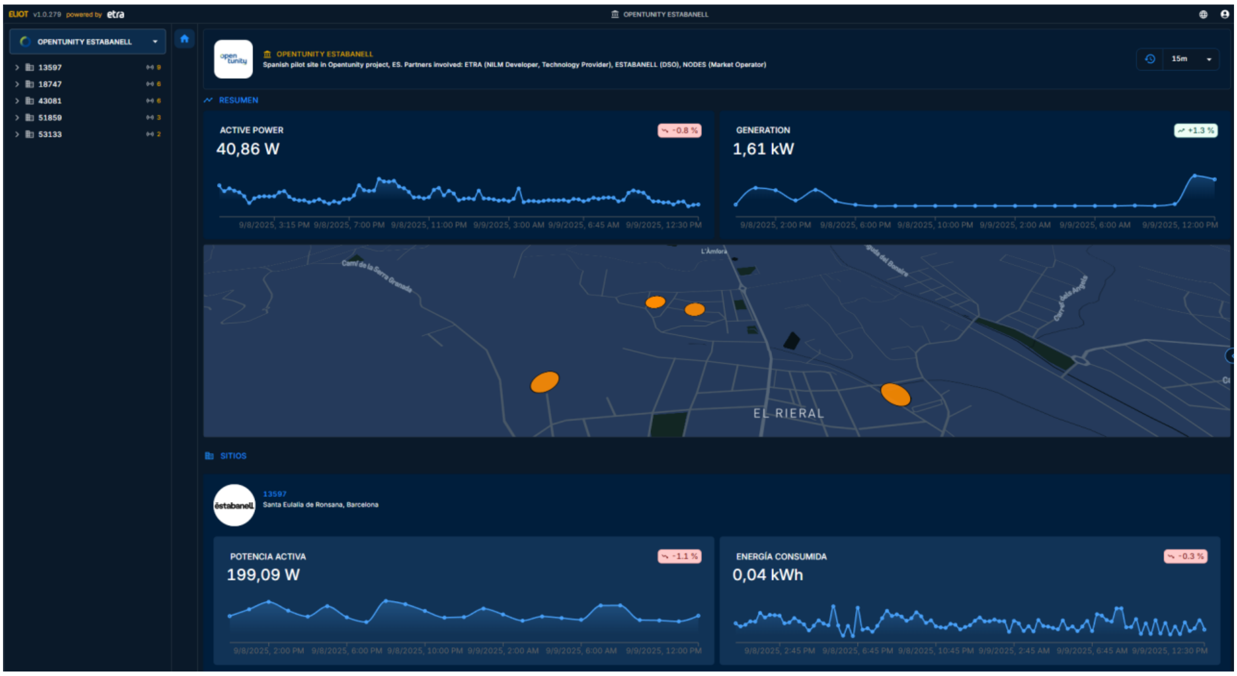


Figure 2: ELIOT Graphical User Interface, dashboard page

A button for more detailed information about the building or household can be selected showing building details page, Figure 3.

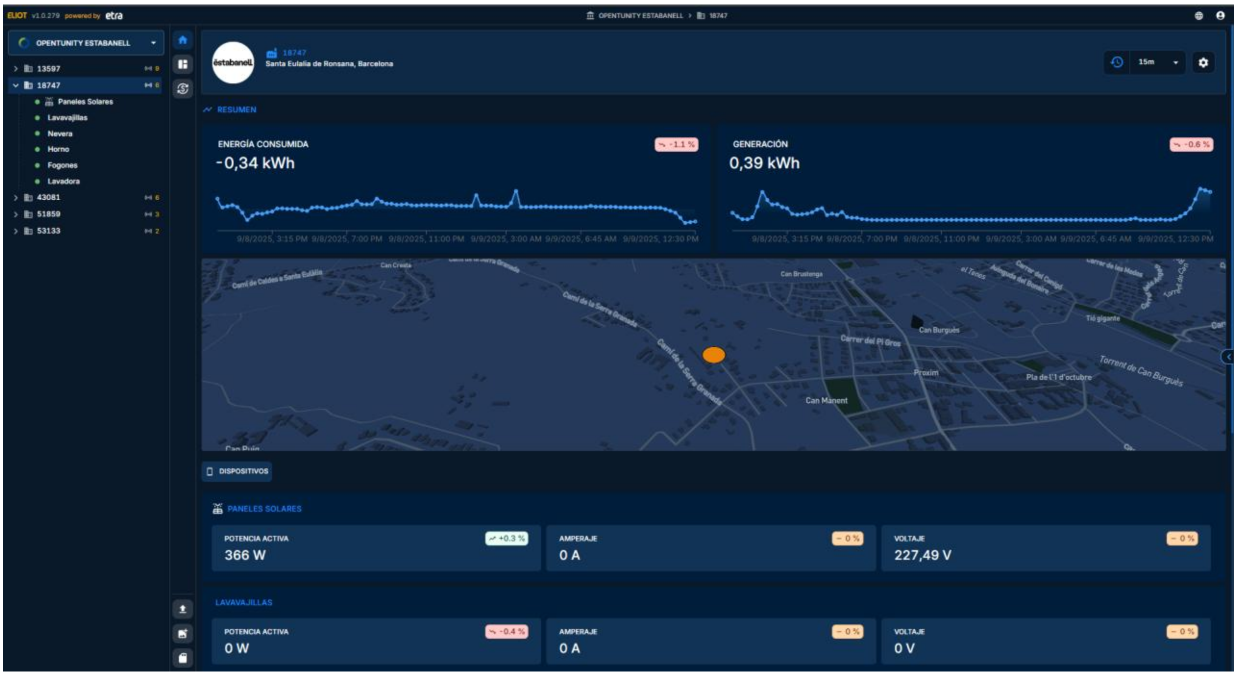


Figure 3: ELIOT Graphical User Interface, building details page

Further detailed information about the devices inside the building can be found, see Figure 4.

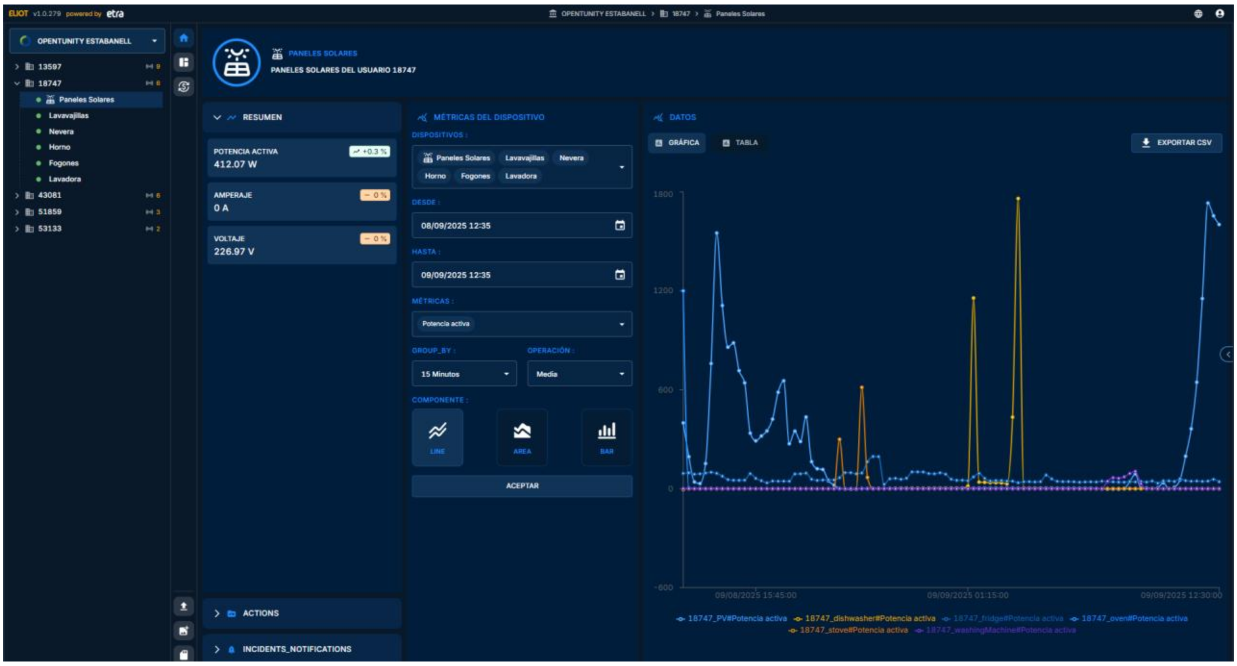


Figure 4: ELIOT Graphical User Interface, asset monitoring page

The data managed by ELIOT is structured hierarchically in four defined levels:

- **Realm or pilot,** a high-level grouping of different installations.
- **Site,** typically represents a building or location where several devices are installed.
- **Zone,** a sub-division of a site (physical or logical) where a subset of devices is located.
- **Device,** a wide range of elements that provide measurements and other information from the field. Devices can be located inside a zone or directly from the site.

In parallel, ELIOT has also been integrated with residential air conditioning devices to enable demand response actions. For this purpose, a dedicated API has been developed, allowing the flexibility service provider to remotely interact with the HVAC units installed in households. This integration has been achieved through the Sensibo API, which enables the modification of setpoint temperatures of AC units connected to a Sensibo smart AC controller, and supports the activation of flexibility offers. Through this new development, ELIOT can both display real-time information and actively send control commands.

2.2 AEM R&D platform - FLEXO

The data from the Swiss pilot site is handled by two different data platforms, the AEM R&D platform and FLEXO, both of which are responsible for data collection, storage and pre-processing. FLEXO has been developed by Hive Power to manage data from the smart meters, while the AEM R&D platform integrates data from all other sensors installed behind the smart meters at building level and on the LV distribution network, in addition to the control signal forwarding which is currently being tested.

Figure 5 shows the general scheme of AEM's data flows from the field devices to the central servers and the web API used to share data outside the organization. AEM's R&D platform, which has been operational since January 2024, enables the R&D team to access and manage company data from the distribution grid, energy communities and buildings for research purposes. Hosted in a dedicated virtual machine on AEM's servers, the platform operates independently of the company's main systems, ensuring that no personal customer data or other types of sensitive data are handled. The platform supports data storage, visualization and secure sharing with authorized third-party users, including OPENTUNITY partners. Internally, AEM is using the platform for asset monitoring and data sharing to third parties in addition to testing its capabilities to enable remote control of specific assets for research activities. Data visualization is based on a dedicated Grafana-powered dashboard, which incorporates an alerting system. Access to the dashboard is restricted to the R&D team. FLEXO on the other hand is cloud-based solution and it is subjected to the same data policy.

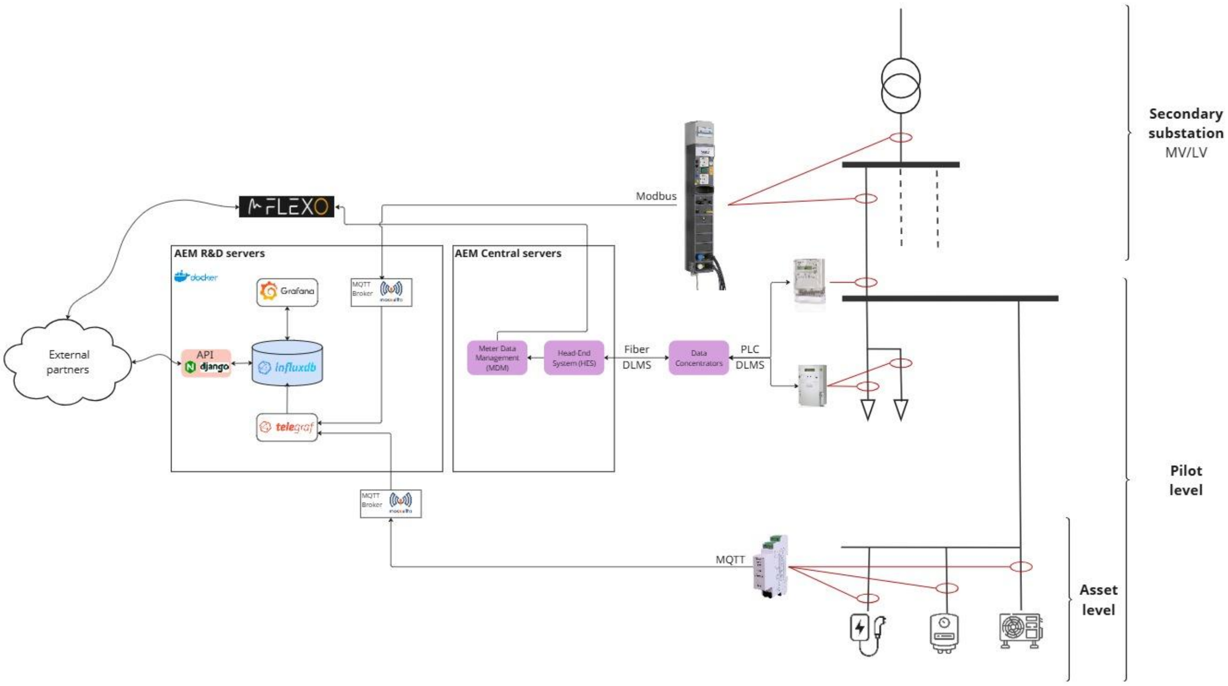


Figure 5: General scheme of AEM's dataflows

The platforms work together for collecting and storing data from all relevant assets and devices installed in the AEM's area of operation, including the energy communities, considering both buildings and the sensors installed in the Low Voltage (LV) grid level. Below is a list of the most relevant sensors:

- Advanced Metering Infrastructure (Landis+Gyr):
  - Type of sensor: smart meter
  - Type of data: net power (W)
  - Data frequency: 15-mins
  - Data updates: 2 hours delays on avg.
  - Asset target: Point of Deliveries (PODs), PV systems, Batteries, EV charging stations
  - Platform: FLEXO
- Submeter (Shelly):
  - Type of sensor: energy meter
  - Type of data: Active power (kW), Apparent power (kVA), Voltage (V), Current (A), Power Factor, Frequency (Hz)
  - Data frequency: 1-min
  - Data updates: quasi real time

  - Asset target: PV systems, Batteries, Heat Pumps, El. boilers, AC systems, EV charging stations
  - Platform: AEM R&D platform
- LV grid monitoring devices:
  - Type of sensor: Smart Grid Interface Modul (SGIM)
  - Type of data: Active power (kW), Apparent power (kVA), Voltage (V), Current (A), Power Factor, Frequency (Hz)
  - Data frequency: 1-min
  - Data updates: quasi real time
  - Platform: AEM R&D platform
- Building sensors:
  - Type of sensor: temperature and humidity
  - Type of data: Temperature (°C), Relative humidity (%)
  - Data frequency: 15-mins
  - Data updates: quasi real time
  - Platform: AEM R&D platform

The communication protocols are variable based on the type of device and the specific manufacturer. Compatibility is currently implemented for protocols like MQTT, OCPP, Modbus and HTTPS. Once collected, in both platforms data undergoes a harmonization process that includes outlier removal.

Despite the two platforms, third-party access is provided using a single web API accessible only through a whitelist of static IPs. Permits are typically limited in time to a specific project duration and a defined list of endpoints. The list includes both smart meter data from the single Point of Delivery (POD) and asset level data and the same API can also be used to control eligible systems.

As mentioned above, the platforms guarantee privacy by not storing or processing personal information, as shown in Figure 5 by the blocks of the 'anonymization process' for the AEM R&D platform, which is instead embedded in the Flexo solution. Sensors are uniquely identified (e.g., ECM46) and all queries and control requests are filtered to guarantee network stability and user comfort. Comprehensive documentation of the API and list of endpoints, including building metadata, is provided to partners and tailored to their specific needs.

Asset remote control has been recently tested internally in AEM and is ready to be provided to external partners for testing their algorithms or solutions developed during the project. These functionalities will vary depending on the type of asset. In the OPENTUNITY project, the Swiss pilot site will focus on the control of small residential assets, including heat pumps and EV charging stations.

EV chargers will be managed directly by AEM, as they are part of the public charging infrastructure and therefore fall outside the scope of BFMS. HYP will be responsible for controlling the heat pumps, with support from AEM. Finally, to assess the impact on a wider range of assets within the pilot site for the flexibility market, the remaining assets, i.e. electric boilers and some heat pumps, will be simulated using an asset simulation tool provided by SUPSI, which will be integrated into the data and control signal pipeline.

Below is an explanation of the control logic per asset type, including operating criteria and examples.

EV chargers

Public EV chargers are connected to AEM's EV management platform, which is based on the Emoti solution. Through this platform, the power profile of each charging point can be retrieved via a GET request, while control actions are performed using a POST request. Control is applied to the power profile (in W) in 15-minute time slots. Both single-slot and multi-slot profiles are supported, with a maximum scheduling horizon of 24 hours, and future requests can be overwritten if needed.

Two important rules must be considered for proper control:

1. Any request that reduces power below 3 kW will stop the charging session.
2. Any request above 11 kW will be limited to that value, which represents the station's maximum charging capacity.

Once a control command has been executed, if a new command is issued, the EV charger will restore the original charging profile according to the vehicle's demand.

Example of command for multi-timestamp:

```
{
    "2025-03-11T12:00:00": 1.0,
    "2025-03-11T12:01:00": -1.0,
    "2025-03-11T12:02:00": 1.0,
    "2025-03-11T12:03:00": -1.0,
    "2025-03-11T12:04:00": 1.0
}
```

Example of test conducted to limit or suspend the charging session:

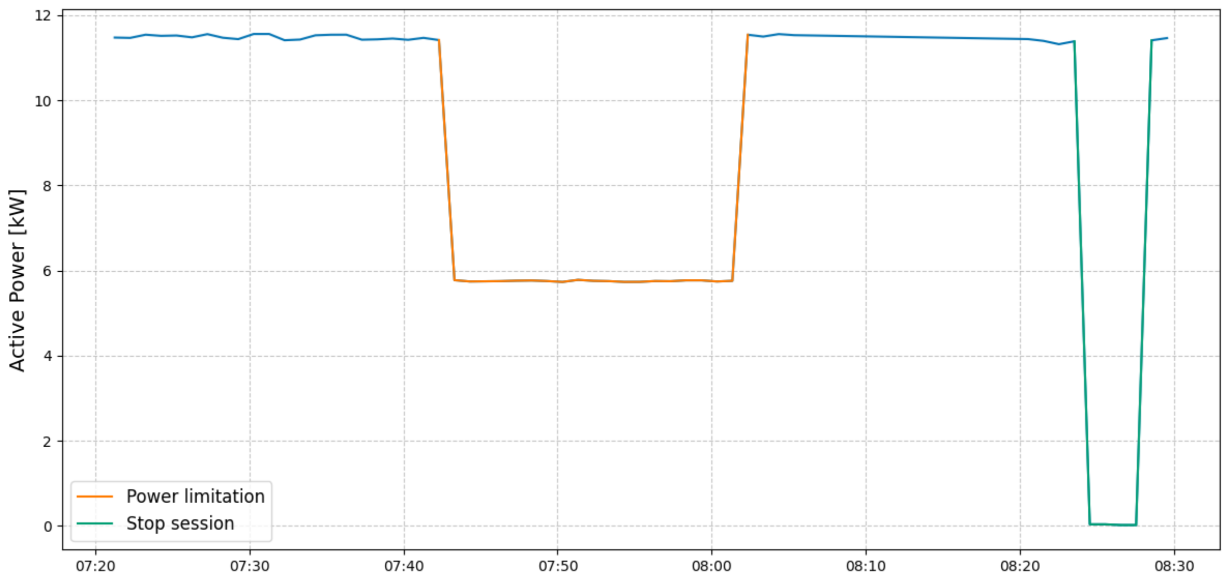


Figure 6 Example of EV charger control test

Heat Pumps (HPs)

The control of heat pumps has been developed for Elco smart grid-ready systems. The available API allows the systems to be turned on or off by sending POST requests. Commands are executed every minute, for example, a command sent at 16:00:35 will be executed at 16:01:00.
Once a command has been sent, it can be overridden by a new one. It's important to note that if no manual commands are received within one hour, the system will automatically switch on and resume operation according to its internal schedule.

Example of command for a single timestamp:

```
{
    "power": false,
    "time": "2025-04-17T16:00:00"
}
```

Example of test conducted to turn off and on the heat pump:

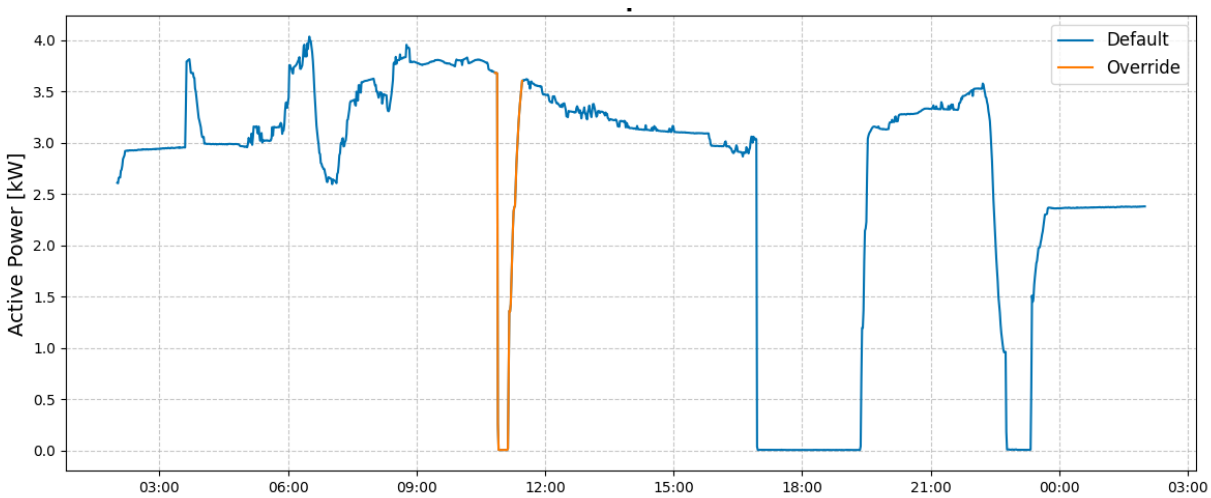


Figure 7 Example of HP control test

2.3 SMART BOX

The HYPERTECH Energy Management System is designed to monitor and control energy consumption and indoor conditions within buildings. It is used at the Greek pilot site of the project. The system involves several actors/components such as Prosumers, who can both produce and consume energy and Spaces, representing specific building areas. Energy loads include Building Global Load, covering total metering, PV production, HVAC and Domestic Hot water (DHW) and Space Load, covering space metering, lighting, sensing, HVAC and other devices. These loads are monitored and controlled through specific and custom-made Monitoring and Control Channels. The system relies on continuous data exchange that is enabled by connected devices, operating through a Wireless Sensor Network (WSN).

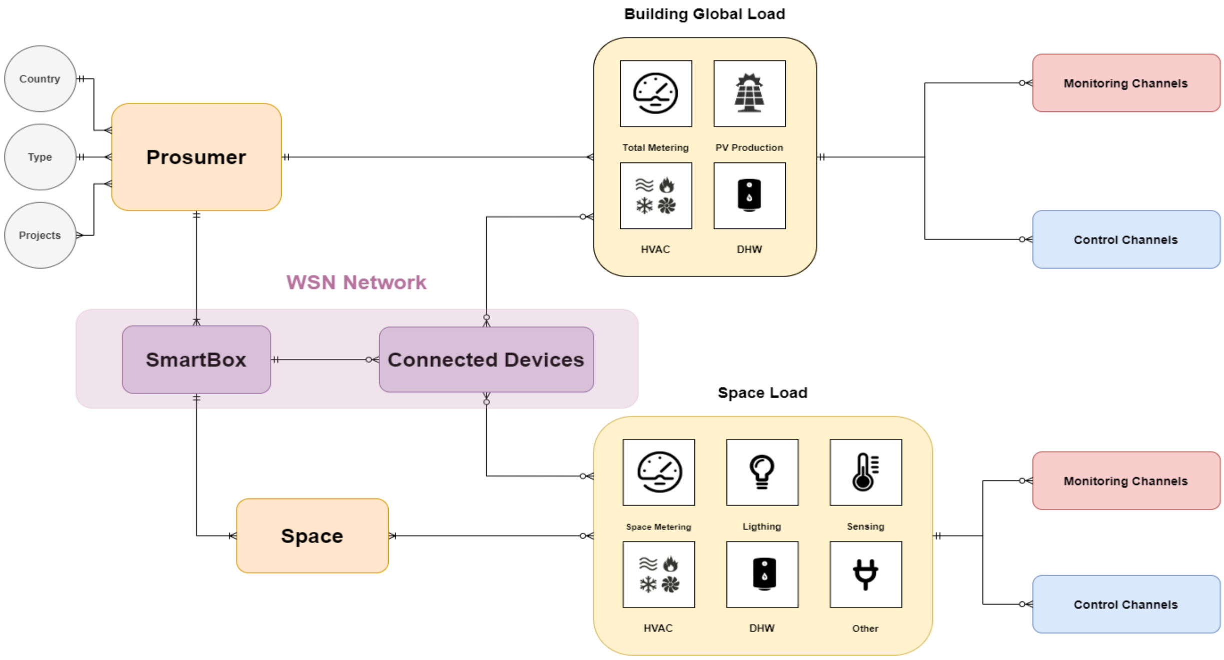


Figure 8: The HYPERTECH Energy Management System

At the center of the system is the **HYPERTECH Smart Box**, developed by HYPERTECH. Its main purpose is to enable seamless, bidirectional communication between a cloud-based system and the various IoT devices installed within the selected dwellings of the project. It serves as a central hub where IoT devices are integrated and managed, therefore enabling wireless and remote monitoring and control of the available flexibility assets. It collects data from various devices, such as indoor conditions, such as temperature, humidity, illuminance, occupancy. In addition, it communicates with controllers and actuators to dispatch control signals, in this way allowing real-time operation following demand response strategies. Moreover, the Smart Box gathers data from metering devices, allowing monitoring energy consumption and appliance performance. The installation of the Smart Box at the pilot sites and its continuous communication with IoT devices transforms the dwellings into smart, energy-responsive buildings, providing necessary data for flexibility forecast and enables the dispatch of flexibility within the OPENTUNITY project.

It is designed under the principles of a Wireless Sensor Network and supports a wide range of IoT devices, including smart meters, sensors and controllers. In this way, real time information on power consumption and indoor ambient conditions from the pilot site is provided to BFMS. The Smart Box is responsible for managing the controllable assets, reporting alerts and facilitating the exchange of information related to demand flexibility. One of its key characteristics is its interoperability, as it is compatible with a wide range of commercially available IoT devices that communicate using standard protocols. In addition, it favors adaptability, modularity and cost efficiency and it is designed to be easy to install and manage while maintaining a user-friendly aesthetic. The whole system consists of smaller, independent components that can be easily managed, implemented and maintained. In this way, complexity is reduced, as each module can be designed, tested and updated separately. Adaptability is also increased, as different modules can be swapped or customized to meet the specific needs of individual users. Modularity simplifies maintenance, making it easier to isolate and address issues and supports future upgrades or changes without impacting the entire system.

The software stack of the Smart Box consists of multiple layers, including a **Network Layer** for seamless connectivity, a **Protocol Agnostic Layer** for the integrating of various IoT protocols into a unified API and an **Application Layer** for device discovery, control dispatching and cloud communication. Moreover, **Security APIs** offer secure data exchange, while **Message Broker** and **REST interfaces** ensure interaction with the cloud infrastructure.

The **Network Layer** is built with a number of key functional specifications, offering reliable and efficient operation. For instance, it operates under low data storage requirements with a minimum capacity of 32GB for local gateways and it prioritizes robustness through regular testing and reconfiguration capabilities. Moreover, it has minimal energy needs, as it features event/threshold-based data exchange (rather than continuous streaming). It can also support large buildings with numerous sensors and actuators, by offering the possibility to use communication extenders or additional gateways. The interoperability feature is ensured via the support of multiple wireless communication protocols, thus, allowing the Smart Box to interface with a variety of sensing, metering and controlling devices, including devices that do not adhere to particular data models. This is particularly interesting for flexibility assets that were not initially intended to have a network interface. The data management model remains extensible while respecting memory constraints of the Smart Box and features short-term data storage to prevent information loss during communication interruptions. The system supports various wireless protocols like Z-Wave, WiFi and BLE, with the possibility to upgrade to additional protocols such as ZigBee and EnOcean.

The **Protocol Agnostic Layer** is a middleware component within the Smart Box, as it creates a bridge between various technologies and the smart building automation system. It manages bidirectional communication by collecting data from IoT devices using different protocols and delivering it to the next layer (application layer); it also translates and sends control commands to the flexibility assets. This is implemented through OpenHAB, an open-source automation software. OpenHAB was selected for its Java based functionality, as it has a vendor-neutral design and offers extensive technological support. Additionally, it is highly adaptable and extensible and can therefore integrate different communication protocols while maintaining a unified interface.

The **Application Layer** is one level higher than the protocol agnostic one. It offers several functionalities, such as IoT device management, data security, control dispatching and system operation maintenance. It consists of several sub-modules, the first one being the Commissioning and Configuration app that enables device discovery and registration. Moreover, a user-friendly Smart Box app allows end-users to control their assets and monitor indoor conditions. The IoT Network Health Check module ensures network integrity by monitoring device status and performing necessary healing processes. Moreover, the Data Management and Backup Module stores data for up to seven-day period, in case there is a cloud connection disruption. The Control Dispatcher Proxy is responsible for the management and execution of control commands. Lastly, the Over-the-air Update Module handles remote firmware updates for core services. The aforementioned modules ensure a reliable system operation while offering remote maintenance capabilities and updates without requiring physical access to pilot sites.

The Smart Box services and applications operate on the Rasberry Pi Operating System, including all necessary drivers. The system's Protocol Agnostic layer is developed under the OpenHAB automation software, using Java extensions to enable communication between the Smart Box and devices at the pilot sites. The Commissioning and Configuration app is developed using Node.js and JavaScript.

The Smart Box uses a Raspberry Pi 3 Model B+ single-board computer, providing a full operating system and development environment (Figure 9). It features a 1.4GHz 64-bit quad-core processor, dual-band wireless LAN, Bluetooth 4.2/BLE, fast Ethernet and Power-over-Ethernet support. Moreover, a RaZberry 2 GPIO daughter card is used to transform the Raspberry Pi into a Z-Wave Home Automation gateway (Figure 9). RaZberry 2 includes a Z-Wave transceiver, a certified Z-Wave communication stack (Z-Way) and a web-based AJAX user interface, allowing easy control and customization via a browser or mobile device, while offering a communication range of up to 200 meters.

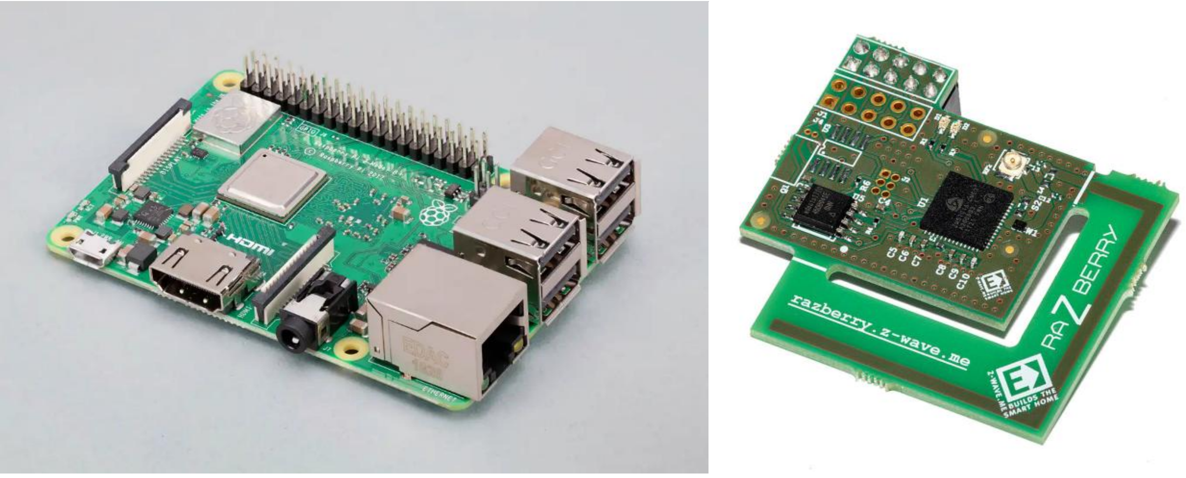


Figure 9: The Raspberry Pi 3 Model B+ (left) and the RaZberry 2 GPIO daughter card (right)

Overall, the Smart Box serves as the central hub that integrates and manages IoT devices deployed in buildings, enabling smart monitoring and automated demand response.

2.4 REDUXI

Reduxi Controller is an innovative energy management system designed to optimize electricity consumption, production and storage for residential and industrial users. Using different optimization modes, the system provides real-time adjustments to reduce costs, enhance energy efficiency and increase self-sufficiency, making it an ideal solution for those looking to align their energy practices with sustainability goals. Its intuitive cloud-based platform offers users complete control and insights into their energy systems, driving smarter decisions and long-term savings.

Reduxi Controller is the HEMS. It's the core component of the Reduxi ecosystem, offering robust functionality to optimize energy management and maximize efficiency. Its advanced capabilities include:

- Centralized Energy Management: Seamlessly connects solar panels, energy storage, heat pumps, EV chargers and more to streamline energy production and consumption.
- Smart Device Integration: Communicates with devices using industry-standard protocols (e.g., Modbus, DLMS, OCPP) for effortless control and compatibility.

- Dynamic Tariff Optimization: Leverages real-time market data to reduce energy costs through automated load shifting and peak management.
- High-Speed Data Processing: Provides second-by-second data acquisition and control for precise and responsive system operations.
- Plug & Play Setup: Simple and user-friendly installation with intuitive configuration tools, enabling fast deployment without technical expertise.
- Scalable Design: Supports an unlimited number of devices for versatile use in residential, commercial, or industrial environments.

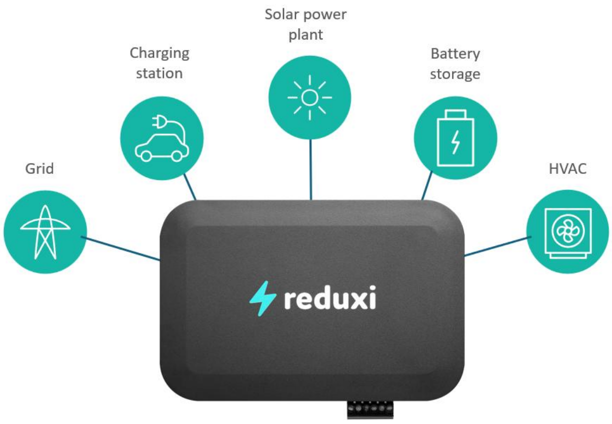


Figure 10: Reduxi Controller connections

Reliable, efficient and highly adaptable, the Reduxi Controller empowers users to achieve energy self-sufficiency and cost savings while staying connected to the future of energy management. The Reduxi Controller represents the central part of the system, connecting with devices via communication protocols, eliminating the need for additional electricity meters or external units to manage connected devices. The Reduxi system allows the integration of solar power plants, battery storage systems, heat pumps and electric charging stations. It automatically manages electricity production and consumption while optimizing grid consumption. Using advanced algorithms, it selects the most cost-effective energy usage path and maximizes self-sufficiency.

The Reduxi Controller is designed for easy installation and use, following the "Plug & Play" approach, requiring no technical knowledge. Installation is completed in just three steps:

1. Power up the Reduxi Controller.
2. Connect it to the internet via wired Ethernet or wireless WiFi.
3. When the green light turns on, the device is correctly connected and installed.

Devices can be connected easily via wired or wireless methods using the Reduxi Configurator with just a few clicks.

Supported Connections

- **Wired Connections:**
  - LAN (TCP/IP)
  - 2x RS485
- **Wireless Connections:**
  - WiFi (TCP/IP)

Figure 11: Reduxi controller and interface

Supported Communication Protocols:

- Modbus RTU
- Modbus TCP/IP
- DLMS
- DLMS-push
- M-bus (EN 13757-2)
- OCPP 1.6J
- REST API
- SunSpec
- MQTT

A full list of supported devices and protocols is available on https://support.reduxi.eu.

The Reduxi Controller can operate independently without the Reduxi Cloud solution or be connected to the Reduxi Cloud for remote access and management anytime, anywhere. Data is transmitted to the Reduxi Cloud in real-time at a second-based interval and periodically every minute. The Reduxi Cloud is accessible via a web browser on desktops and mobile devices and is free for end users. With advanced encryption, the Reduxi Cloud ensures high security for both data and connected electrical devices, complying with GDPR regulations.

The Reduxi Gateway expands the Reduxi Controller system by converting RS485 serial signals into internet-based TCP/IP communication. It is useful when additional serial inputs are needed or when the connection distance is too long for a wired connection. The Reduxi Gateway also enables third-party device control via two relay outputs, such as controlling a heat pump using SmartGrid Ready (SG Ready) or EVU contacts.

Each Reduxi Controller comes pre-installed with the Reduxi Configurator, accessible via a local network browser or the Reduxi Cloud. Users can easily add devices and configure strategies through drop-down menus. Using the MQTT protocol, data can also be accessed directly through a private or public MQTT broker. The MQTT API is fully open and publicly available at https://mqtt.support.reduxi.eu.

The Reduxi system supports real-time energy meter data collection through P1 port of smart meter. Users can connect energy meters and obtain billing data every second without additional power supply requirements. The Reduxi Controller maximizes energy self-sufficiency by prioritizing self-generated solar energy and battery storage over grid consumption. It adjusts energy use based on market prices, ensuring cost-effective operation by charging storage systems and EVs during low-cost periods and discharging during high-cost periods.

All functionalities and strategies are executed locally, meaning that even if the internet connection is lost, the Reduxi Controller continues to operate effectively. The Reduxi Controller offers public documentation for MQTT API integration, supporting both public and private MQTT brokers (AWS, Azure, Google Cloud, etc.). Users can customize data transmission frequency and use publish-subscribe methods for real-time device control.

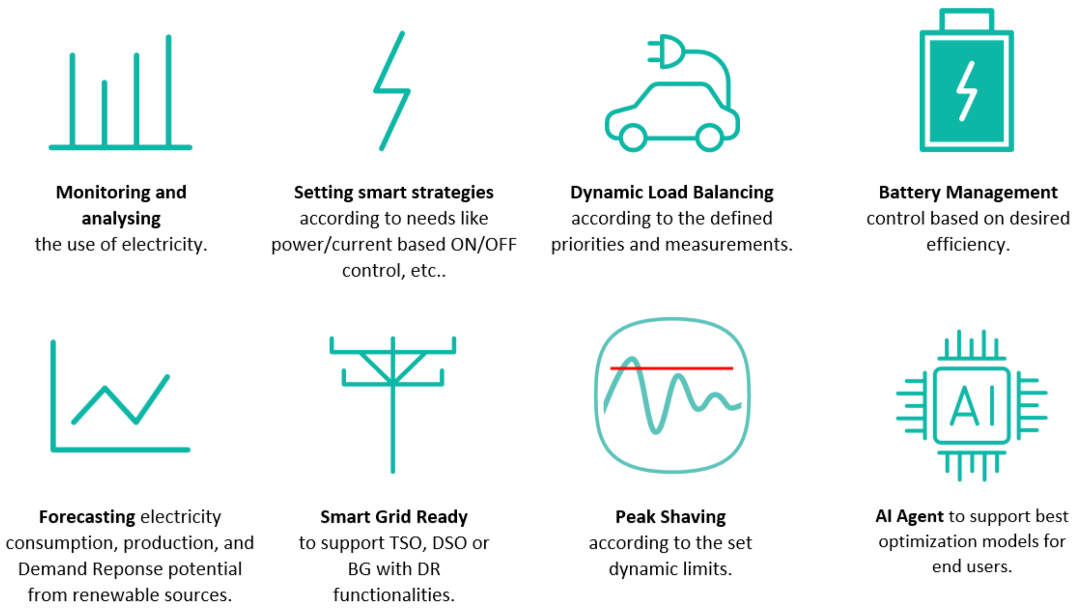


Figure 12: Reduxi capabilities

3. Building Flexibility Management System

3.1 Introduction

3.1.1 Scope of component

The Building Flexibility Management System (BFMS) is one of the innovative tools of the OPENTUNITY ecosystem, designed to forecast and dispatch building- and asset-level flexibility. It ensures that flexibility assets are accurately identified, quantified and activated without compromising end-user comfort or the operational integrity of building systems. Its scope covers the full chain of flexibility delivery, starting with the collection of consumption data and occupant preferences, proceeding with the forecast of baseline and flexibility values and extending to the execution of DR events and market participation.

BFMS development is based on the requirements of two Demand Response Use Cases of the project, i.e., Use Case 1.8 "HEMS/BEMS DR optimization and local flexibility management" and Use Case 1.9 "Initialization of HEMS/BEMS Demand Response strategy". A detailed description of the Use Cases of OPENTUNITY can be found in Deliverable 2.3 "Open architecture report".

3.1.2 Purpose and Functionality

The purpose of BFMS is to transform buildings into active and reliable participants in energy flexibility schemes. It addresses the dual challenge of ensuring efficient grid interaction and maintaining occupant satisfaction. At its core, BFMS provides the means to optimize flexibility at the building level, aligning local consumption with external DR signals while offering tangible value to both end-users and system operators.

Functionally, BFMS is organized around a modular architecture and consists of different Services (Figure 13). The DR Initialization Service enables users to define preferences and operational boundaries. The Profiling and Flexibility Forecasting Orchestration Service captures necessary information on comfort levels, building thermal behavior and asset consumption patterns to generate day-ahead baseline profiles and flexibility estimates. The Control Dispatching Service translates these forecasts into control signals dispatched at the pilot sites of the project. Lastly, the Flexibility Service Provider even though not a direct part of BFMS, is an associated and necessary service, as it aggregates the forecasted flexibility and proceeds with the bidding process at the flexibility market. Together, these Services ensure a seamless workflow from data acquisition to flexibility dispatch. A detailed presentation of each Service is provided in the following sections.

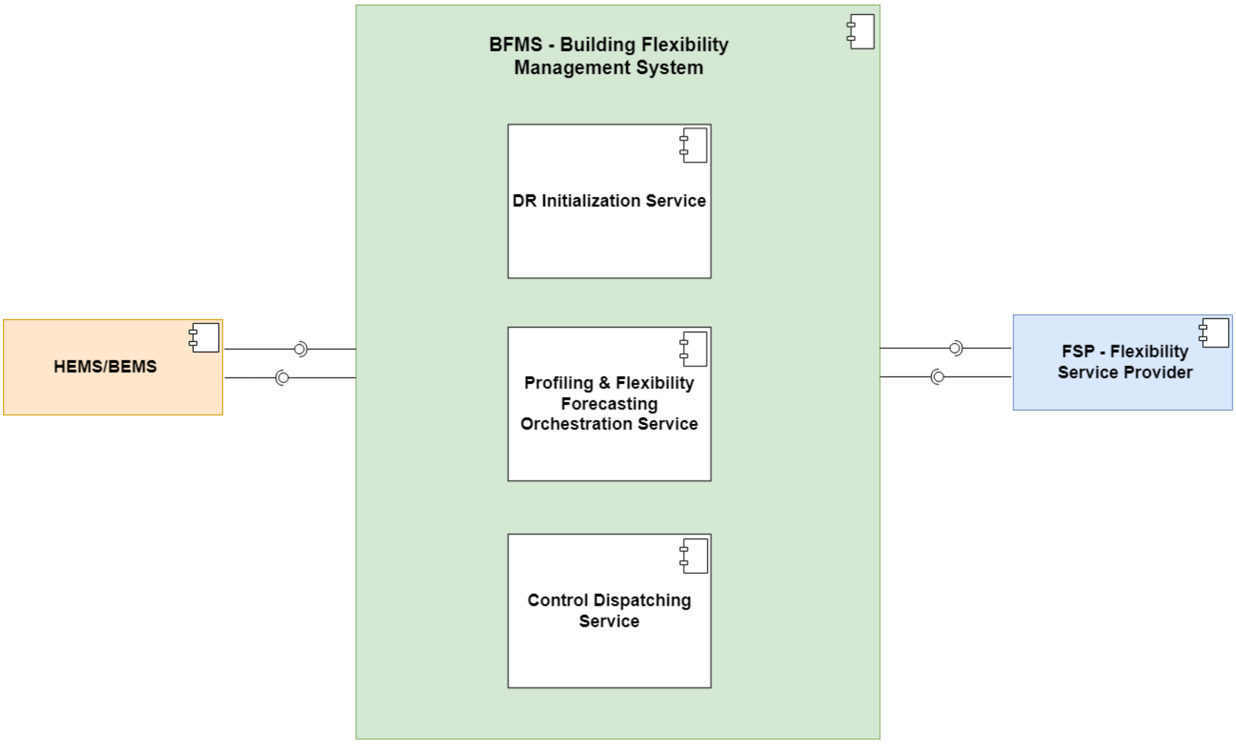


Figure 13: BFMS component diagram

3.1.3 Interactions and Integration

BFMS achieves its objectives through close interaction with other systems, services and stakeholders within the OPENTUNITY framework (Figure 14). At the building level, it interfaces with HEMS/BEMS platforms (as described in section 2 OPENTUNITY HEMS/BEM), with BFMS receiving energy consumption data from the HEMS/BEMS in the form of power time series. Upon availability, HEMS/BEMS also provide sensorial data (temperature, occupancy, etc.) and/or occupant defined preferences/constraints, with the implication of the DR Initialization Service. BFMS consists of a number of modules, that will pre-process this data, proceed with comfort, building and asset profiling and ultimately calculate the baseline and flexibility power consumption. The forecasted baseline and flexibility values are then sent to the Flexibility Service Provider, which performs an evaluation of the available aggregate flexibility from the available assets. At the market level, the FSP handles the flexibility bidding to the NODES platform. System operators submit their buy orders on the NODES platform and once flexibility offers are accepted, the corresponding offers return to the FSP. The FSP then communicates this information back to BFMS, which will in turn assess the feasibility of flexibility dispatch based on real time information from the assets. Appropriate control command sequences are then generated and sent to HEMS/BEMS.

Through these interactions, BFMS ensures interoperability across diverse environments and stakeholders, establishing a coherent link between residential buildings, flexibility aggregators and market platforms. The integration guarantees that flexibility delivery is not only technically feasible but also economically viable and user centric.

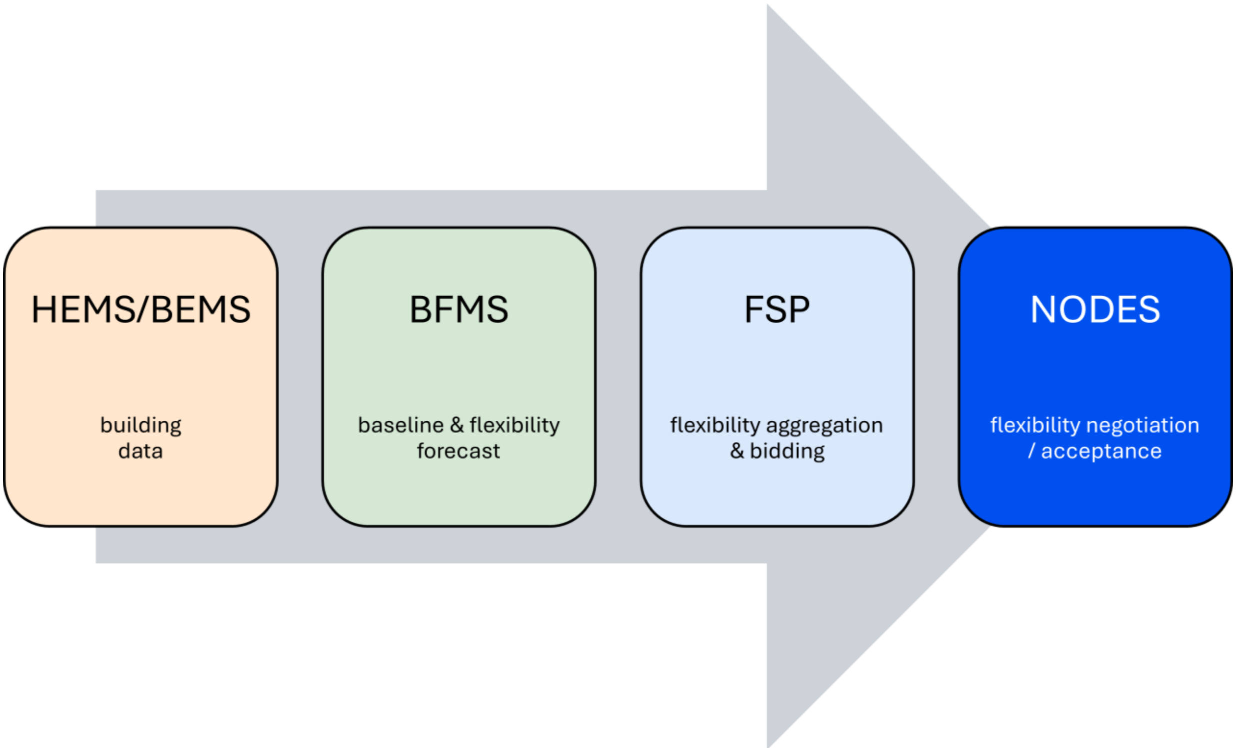


Figure 14: High-level interaction diagram between BFMS and other services of the project

3.2 Additional requirements posed by the OPENTUNITY ecosystems

BFMS operation is based on components that have been developed in past EU funded projects. However, within the context of OPENTUNITY new requirements emerged that necessitated both functional and structural adaptations. The new requirements were identified through a detailed process, taking into account the specificities of the pilot sites, the need for interoperability, as well as the previously defined Use Cases and respective requirements (as presented in Deliverable 2.1 Technical foundations). The additional requirements posed by the OPENTUNITY ecosystem are detailed in this section.

One of the main challenges comes from the diversity of HEMS/BEMS solutions deployed across the pilots. Unlike the homogeneous environment of HYPERTECH's original platform, the pilots rely on heterogeneous systems that cannot be easily adapted to a single common model.

BFMS must support a broad range of functional and non-functional requirements. From a functional perspective, these include adjustments in communication, semantics, syntactic structures, user preferences inclusion, as well as baseline and flexibility algorithms. Communication-wise, while the original BFMS employed specific HTTP REST endpoints and brokers, it must now accommodate additional communication pathways. Semantically, the system was tightly bound to HYPERTECH's use cases and respective data models, while for the OPENTUNITY project it must handle the varying semantics and data models of each of the new pilot's HEMS/BEMS. Syntactically, messages previously tailored to a single platform must now be defined in formally specified formats that allow interoperability across diverse external ecosystems.

The management of occupants' preferences is another important element of change. Previously, user preferences were implicitly inferred through asset utilization profiles. In the OPENTUNITY context, the DR Initialization Service allows preferences to be explicitly defined by users, enabling them for instance to select or deselect devices for DR participation and to specify time windows for interventions.

In addition, baseline and flexibility algorithms concerned a fixed set of models. Instead, BFMS must support the deployment of new or additional algorithms. Similarly, the Control Dispatching Service was originally implemented for a specific platform, whereas now it must comply with the diverse communication, semantic and syntactic approaches of multiple ecosystems.

In terms of non-functional requirements, adaptations are needed performance-, privacy- and security-wise. Performance considerations demand scalability to multiple platforms, supporting a much larger user pool while maintaining efficient resource use. Privacy must be preserved across distinct platforms, ensuring strict separation of data, consent and policy enforcement. Likewise, security requirements evolve from a single-realm context to managing multiple realms and security frameworks in parallel.

In addition to these elements, the integration of BFMS into OPENTUNITY's Data Space imposes new demands in terms of infrastructure, interfacing and governance. Furthermore, new functionalities must be incorporated, including flexibility aggregation modules capable of identifying and consolidating flexibility potential. Finally, integration with external market platforms, i.e., the NODES platform, requires the development of new middleware / clients and interfaces that can support combined aggregation and bidding processes.

In summary, the additional requirements imposed by the OPENTUNITY ecosystems necessitate enhancement of BFMS from a single-platform solution into an interoperable and scalable system, fully capable of operating across heterogeneous pilots, data infrastructures and market environments.

3.3 BFMS ecosystem

This section presents the methodologies that have been employed for the core Services of BFMS component. It is divided into the following subsections:

- Profiling and Flexibility Forecasting Orchestration Service (core service)
    - Profiling Modules:
        - Comfort Profiling Module;
        - Building Thermal Modeling Module;
        - Asset Profiling Modules;
    - Flexibility Forecasting Module;
    - Orchestration Modules;
- DR Initialization Service;
- Control Dispatching Service;
- Flexibility Service Provider.

The main innovation presented in this deliverable is the Flexibility Forecasting Module, that implements an optimization problem which is presented in detail in Section 3.3.2.4. For the sake of completeness, modules that have been developed by HYPERTECH in past EU funded projects are briefly presented (H2020 ACCEPT project [1]).

In previous versions of the Flexibility Forecasting Module, the aforementioned optimization problem was based on the outputs of the Comfort Profiling, Building Thermal Modeling and Asset Profiling Modules. However, this approach required extensive sensorial data, leading to an expensive, complex and intrusive solution. Additionally, unless the sensorial and metering devices were properly installed and maintained, the previous approach could lead to biased data and consequently inaccurate model training. Therefore, the Flexibility Forecasting Module has been refactored to address these drawbacks by introducing a new optimization problem. This problem is formulated using only assets' power profiling data and constraints imposed using analysis of this data (i.e., inferred constraints) or occupants' input via the DR Initialization Service (section 3.3.3) (i.e., explicit constraints).

The Control Dispatching Service and the Flexibility Service Provider are presented in sections 3.3.4 and 3.3.5 respectively.

3.3.1 Conceptual architecture

As already mentioned, BFMS is composed of several Services and Modules, Figure 15 providing its component diagram. The core of the system consists of the DR Initialization Service, the Profiling Modules, the Flexibility Forecasting Module and the Control Dispatching Service. These modules and services are described in detail in the following subsections. Several other modules/services ensure the orchestration of the different procedures, as well as data processing and storage.

The DR Initialization Service serves as the entry point, where occupants interact with the system through a front-end UI module. A back-end module manages occupants' assets and preferences, ensuring that flexibility settings align with user-defined preferences and constraints.

Within the Profiling & Flexibility Forecasting Orchestration Service, several modules collaborate to handle scheduling, data ingestion and execution planning. Profiling modules include Comfort Profiling, Building Thermal Modeling and Asset Profiling Modules, all of which serve as inputs to the flexibility forecasting module. This module is responsible for the forecast of day-ahead baseline and flexibility power consumption timeseries.

For Control Dispatching, a control design module is used, alongside a dispatcher module, ensuring that optimized flexibility strategies are executed efficiently.

The Profiling Storage & Retrieval Service maintains and retrieves profiling and forecasted flexibility results through a persistence service and an external API, allowing data access and integration with external tools and services of the project.

The figure also illustrates the interactions between HEMS/BEMS and BFMS, where building and asset data from the pilot sites, along with control dispatch signals are exchanged. Furthermore, information on the extracted flexibility and demand response control signals are exchanged with the Flexibility Service Provider, ensuring the coordination within the energy market.

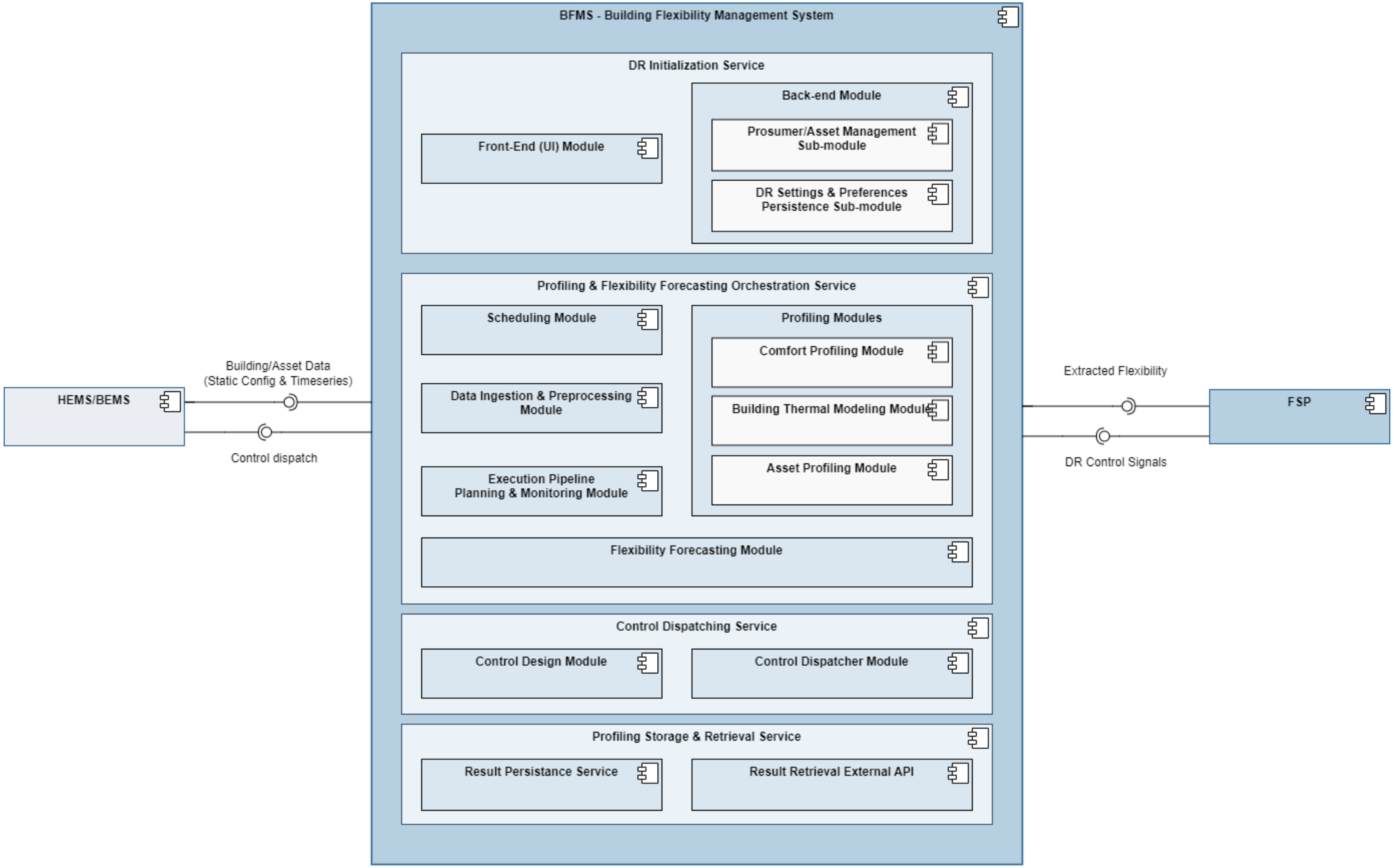


Figure 15: BFMS Component Diagram

3.3.2 Profiling and Flexibility Forecasting Orchestration Service (core service)

The Profiling & Flexibility Forecasting Orchestration Service is responsible for the coordination of the operation of a number of services related to the building performance representation and its flexibility forecast. The service includes a Data Ingestion & Preprocessing Module to gather and process building and asset data, an Execution Pipeline Planning & Monitoring Module to ensure the processes are executed efficiently and in the correct sequence and a Scheduling Module to program the needed processes.

The core operation is performed by a number of specialized profiling modules. The Comfort Profiling Module analyzes data to understand occupant comfort preferences; the Building Thermal Modeling Module characterizes the thermal behavior of the building; lastly, the Asset Profiling Module creates digital models of individual energy assets like HVAC systems and water heaters. The outputs from all these profiling modules are then provided to the Flexibility Forecasting Module. This last component analyses the comfort constraints, building thermal behavior and asset performance to calculate the building's day-ahead baseline energy consumption and its available upward and downward flexibility. These Modules are presented in detail below.

3.3.2.1 Comfort Profiling Module

The Comfort Profiling Module builds on previous European projects like ACCEPT, combining the Random Forest (RF) algorithm with Symbolic Aggregate Approximation (SAX). RF is a commonly used machine learning algorithm that combines the output of multiple decision trees to reach a single result. Its ease of use and flexibility have fueled its adoption, as it handles both classification and regression problems. SAX is another widely used technique for data compression and noise reduction in timeseries data. It simplifies complex datasets by converting timeseries chunks into strings of symbols, making pattern recognition more efficient.

Here, SAX is used to categorize comfort states, while RF is applied for training and predicting them. The integration of these two techniques enhances the ability of the Comfort Profiling Module to analyze and predict user comfort states, increasing the accuracy and responsiveness of BFMS. As previously mentioned, the Flexibility Forecasting Module can use the outputs of the Comfort Profiling Module for the calculation of the baseline and flexibility energy consumption. The Comfort Profiling Module works in tandem with the DR Initialization Service, together offering a human-centric solution that captures the preferences of the occupants.

3.3.2.2 Building Thermal Modeling Module

Building upon the insights from the H2020 ACCEPT project [4], the thermal behavior of buildings in the OPENTUNITY project relies on a data-driven, grey-box modeling approach. It uses a second-order lumped element Resistance-Capacitance (RC) network, leveraging a state-space formulation to model the thermal dynamics of each building space based on measurable inputs such as outdoor temperature, solar irradiance, HVAC loads and internal heat gains. In this way, prior knowledge of the characteristics of the building is not required.

Once parameters are identified, space air temperature predictions are obtained using an analytical solution of the state-space model. This method ensures a balance between accuracy and computational efficiency, making it a viable alternative to conventional white-box thermal modeling methods.

3.3.2.3 Asset Profiling Module

The Asset Profiling Module is responsible for the creation of digital twins of individual energy assets within a building. It transforms data from the HEMS/BEMS into predictive models that describe how each asset behaves under different conditions.

The process begins with Input Data Processing, where the module collects and processes time-series data (e.g., power consumption, temperature, occupancy, control settings) received from the HEMS/BEMS (Figure 16). This data is standardized and validated for consistency and use across other tools and services.

Following that, the Asset Model Training applies machine learning techniques to create predictive models for each asset type (e.g., HVAC, EWH). These models capture the relationships between energy consumption, environmental variables and control actions.

The Asset Forecast sub-module uses these trained models to predict future asset behavior over predefined periods, usually day-ahead. These forecasts help estimate the available flexibility potential, as well as the operational limits of each asset.

In the last step, the Output Data Post-Processing sub-module formats, aggregates and transmits the forecast results to other BFMS components and to external actors (e.g., the FSP).

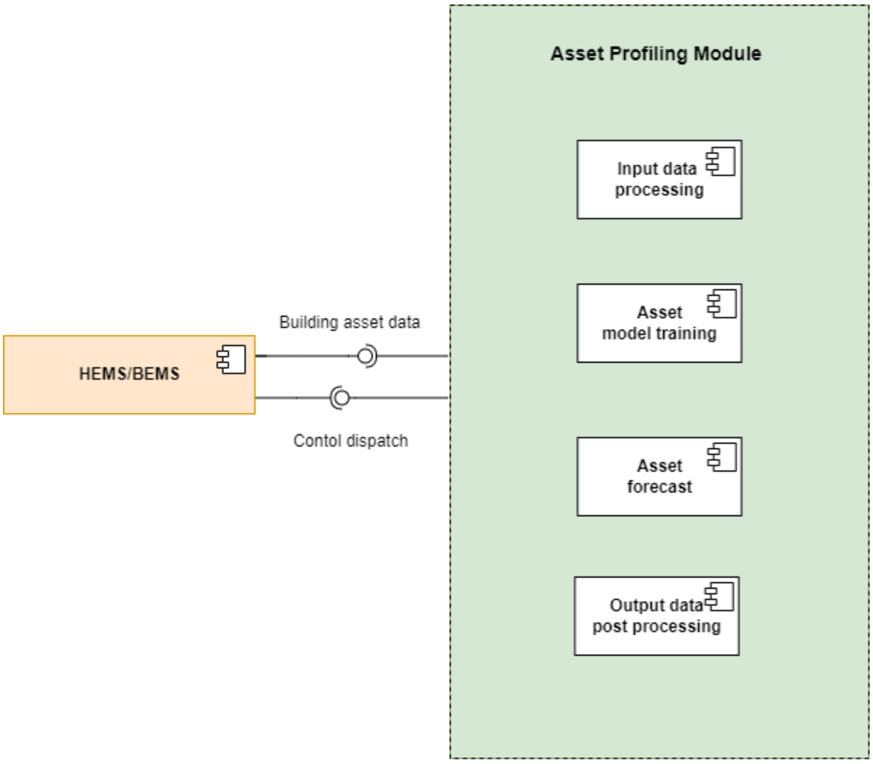


Figure 16: Component diagram of the Asset Profiling Module

This following subsections describe the data-driven methods adopted in OPENTUNITY for modeling HVAC systems (AC split units with inverters, air-to-air constant volume heat pumps and electric heaters, air-to-water and water-to-water heat pumps) and Electric Water Heaters (EWHs) based on ACCEPT D4.8 Building Digital Twin v2 [4]. These models focus on identified types of devices present at pilot sites, ensuring relevance and applicability to real-world scenarios.

For each case, IoT data undergo preprocessing, including outlier detection and removal. This ensures clean and reliable data for model development without requiring further segregation for performance assessment.

3.3.2.3.1. AC Split Units with Inverters

Due to their dynamic behavior, which is influenced by various external parameters such as weather conditions and thermostat settings, advanced machine learning algorithms are used. Specifically, Gaussian Processes (GP) were selected for their non-parametric nature, enabling flexible and scalable modeling as the dataset grows. Unlike parametric methods, which require manual configuration, GP models adapt automatically to varying system specifications.

The GP-based model forecasts power consumption at each time step, providing a range of possible values (mean, minimum, maximum) based on input parameters. The model can be further extended to support more complex HVAC configurations as needed.

3.3.2.3.2. Air-to-Air Constant Volume Heat Pumps and Electric Heaters

Building on methodologies explored in prior research, two clustering-based, data-driven approaches have been identified for modeling air-to-air constant volume heat pumps: DBSCAN [5] and k-means [6]. While DBSCAN can produce clusters with lower variability, it has higher computational demands and requires additional preprocessing steps for hyperparameter definition, making it less practical for this context.

The k-means algorithm, widely recognized for its simplicity and efficiency, was selected as the preferred approach. It is an unsupervised machine-learning method that groups data points into clusters based on their similarity, measured using Euclidean distance. The algorithm requires specifying the number of clusters (k), which is determined automatically using methods like the Elbow and Silhouette coefficient analysis [7]. Once initialized, k-means iteratively assigns data points to clusters, recalculates cluster centroids and converges when no significant changes occur. The enhanced version, k-means++ [6], improves performance by optimizing the selection of initial centroids.

The goal of k-means in this application is to cluster the power consumption patterns of HVAC devices based on available data. Key inputs include the device's power consumption (timeseries data), operational status (on/off) and user-set temperature setpoints. These parameters enable the extraction of clusters that represent different levels of power consumption.

3.3.2.3.3. Air-to-Water and Water-to-Water Heat Pumps

For the profiling of air-to-water and water-to-water heat pumps, a machine learning method is used [2]. The forecasting process consists of multiple steps: data pre-processing, feature extraction, algorithm selection, fine-tuning model parameters and validation using performance metrics.

To improve accuracy, the Extreme Gradient Boosting machine learning technique is used, building multiple decision trees iteratively. Hyperparameter tuning is conducted using the Optuna framework to enhance model performance. Additionally, model stacking is used as an ensemble method, combining multiple models to improve forecasting accuracy.

This approach provides accurate predictions to support efficient operation and management of air-to-water and water-to-water heat pumps.

3.3.2.3.4. Electric Water Heaters (EWHs) Model

EWH modeling focuses on non-intrusive solutions using power consumption and on/off status data to recognize patterns in energy use [2]. The process begins with identifying different Power Consumption Levels (PCLs). Among them, four distinct levels are established, but the lowest one—representing standby or off mode—is set aside at this stage. The remaining data is then analyzed to create a matrix that tracks how frequently each power level appears at different times throughout the day.

Once this occurrence matrix is built, the next step is pinpointing the key time intervals when each power level is most likely to occur. By focusing on the moments when each level appears most frequently, a clearer picture of the heater's daily usage pattern emerges. Finally, this information is used to generate a forecast of power consumption for the following day.

This method provides a practical way to predict how EWHs will behave without requiring complex sensors or intrusive data collection, making it well-suited for IoT applications and flexibility forecasting purposes.

3.3.2.4 Flexibility Forecasting Module

The Flexibility Forecasting Module is one of the core components of BFMS (Figure 17). It estimates the building's flexibility potential, i.e., its ability to shift or adjust energy consumption without compromising comfort or operations. It uses inputs from the Profiling Modules, i.e., the Comfort, Building Thermal Modeling and Asset Profiling Modules, acquiring in this way all necessary information for the flexibility forecast.

The module can be divided into the following processes:

- Input Data Processing: The module begins by receiving and processing the necessary input data. The two key inputs are the baseline energy consumption (i.e., the forecasted business-as-usual energy usage of the building) and optionally, if available, the day-ahead electricity prices.
- Flexibility Forecasting: Using the processed input data, this sub-module performs its core calculations to determine how much and when the building's energy consumption can be adjusted. This involves solving an optimization problem that finds the most economically

beneficial way to shift energy use, resulting in a forecast of available upward and downward flexibility.

- Output Data Post-Processing: Lastly, results from the forecasting engine are formatted and prepared for further use by other services and tools.

The following subsections present the approach used for flexibility forecasting, which represents a key advancement in the project's methodology. The development of the flexibility forecasting module primarily relates to T4.1, with overlaps in T4.2, where specific optimization constraints are identified. User preferences and requirements play an integral role in shaping these constraints, with input collected directly from users via forms containing targeted questions.

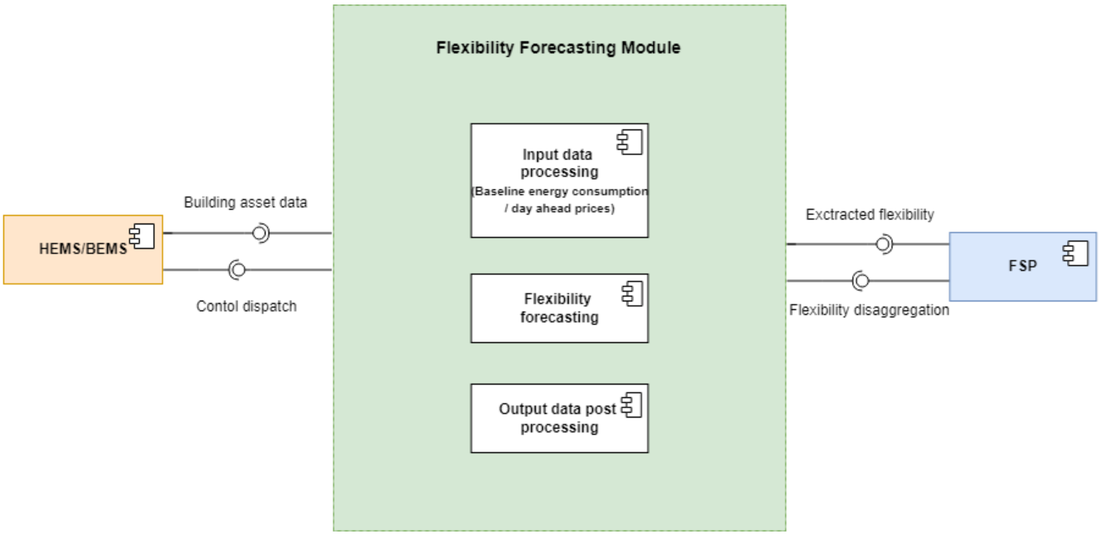


Figure 17: Component diagram of the Flexibility Forecasting Module

Specifically, in the following an update on the problem formulation of the optimization problem related to flexibility forecasting is provided. A cost minimization problem formulation is now employed to take account of electricity prices, if available. It should be noted that load can only be shifted to earlier steps and when price (if available) is lower. More specifically, the end time step of a flexibility profile activation is at most equal to the starting time of the corresponding activation of the baseline profile. The full problem formulation (i.e., decision variables, constraints, objective function) is given below. Finally, it should be noted that the aim of this optimization problem is to provide demand response (DR) while also minimizing cost. Therefore, priority is given to demand response shifting even if prices at this time interval are higher than during the baseline time interval. This is an intentional choice as the economic benefit of participating in DR events greatly outperforms any cost-minimization shifting.

3.3.2.4.1. Decision Variables

Firstly, a separate demand variable $dx_{i,j,t}$ for each asset i (ϵ 1,...,N) is defined, for each activation j (ϵ 1,...,$\alpha[i]$), for each time step t (ϵ 1,...,T), where N is the number of demand assets, $\alpha[i]$ is the number of activations of asset i, and T is the number of time steps. Secondly, $y_{i,j,h}$ binary variables are defined, which are used to choose one of some possible time intervals during which an activation can take place. A y variable for each asset i, for each activation j is defined, and each possible time interval h (ϵ 0,...,F). F is the flexibility window in time steps. h = F corresponds to the baseline. h = 0 represents a demand shifting F time steps earlier compared to the baseline profile. Finally, d_t, is defined, representing the total demand at each time step.

3.3.2.4.2. Constraints

This subsection presents the constraints of the model. A new constraint has been added to the existing ones, which is presented below:

$$if\ dur_{i,j} > flex,\ y_{i,j,F} = 1 \tag{1a}$$

Constraint (1a) enforces the flexibility profile to be equal to the baseline, in case the duration of an activation is greater than the flexibility window (*flex*).

$$if\ dur_{i,j} < flex,\ y_{i,j,h} = 0,\ h \in \left[flex - dur_{i,j} + 1, F - 1\right] \tag{1b}$$

Constraint (1b) excludes intervals that overlap with the baseline profile. This ensures that the end time step of a flexibility profile activation is at most equal to the starting time of the corresponding activation of the baseline profile.

Subsequently, the variable bounds are presented, followed by the rest of the model constraints.

$$dx_{i,j,t} \leq \mathrm{mean\_power}_{i,j} \tag{2}$$

$$y_{i,j,h} \in \{0,1\} \tag{3}$$

$$d_t \geq 0 \tag{4}$$

where mean_power$_{i,j}$ is the mean power during activation j for asset i. It should be noted that given the baseline profile for an asset, the demand over each activation period is averaged at first, and then this averaged demand in the optimization problem is considered. Equation (2) denotes that $y_{i,j,h}$ variables are binary, which means that they can either take the value of one or zero. Equation (4) ensures variable d_t is nonnegative.

Constraint (4) enforces $dx_{i,j,t}$ to be zero outside of all possible activation time intervals.

$$dx_{i,j,t} = 0,\ t \notin \left\{t\_s_{i,j} - F + 1, t\_e_{i,j}\right\} \tag{5}$$

where $t\_s_{i,j}$ and $t\_e_{i,j}$ are the start and end times for asset i and activation j, respectively.

Constraint (6) ensures that the optimized consumption profile has the same amount of energy as the original, for each activation j for each asset i.

$$\sum_t dx_{i,j,t} = \mathrm{dur}_{i,j} \cdot \mathrm{mean\_power}_{i,j} \tag{6}$$

where dur$_{i,j}$ is the duration of activation j for asset i and is equal to $t\_e_{i,j}$ - $t\_s_{i,j}$ + 1.

Constraint (7) provides the relationship between the total demand d_t at each time step and the separate demand variables $dx_{i,j,t}$ for each activation for each asset.

$$\sum_i \sum_j dx_{i,j,t} = d_t \tag{7}$$

Constraints (8) and (9) ensure that one of the F possible time intervals is chosen for each activation for each asset. Constraint (7) can be informed from user preferences that will be captured through T4.2.

$$\sum_{t=t\_s_{i,j}-F+1+h}^{t=t\_s_{i,j}-F+1+h+\mathrm{dur}_{i,j}} dx_{i,j,t} \geq \mathrm{dur}_{i,j} \cdot \mathrm{mean\_power}_{i,j} \cdot y_{i,j,h} \tag{8}$$

$$\sum_{h=1}^{h=F} y_{i,j,h} = 1 \tag{9}$$

Finally, a given amount of time between consecutive activations is ensured. Specifically, this time (as a default option) is set to be one hour for HVAC and three hours for DHW (domestic hot water DHW and electric water heater EWH terms are used interchangeably). This is performed by enforcing specific y variables to be zero, while also ensuring that at least one activation is possible.

3.3.2.4.3. Objective Function

The objective function is:

$$\min \sum_t p_t d_t \tag{10}$$

which incentivizes demand d_t to be increased when price p_t is low, and decreased when price is high.

Table 1 summarizes assumptions for the optimization problem. Finally, Figure 18 presents a flowchart which shows the steps of the overall process.

Table 1: Summary of assumptions for the optimization problem

Item	Explanation
Concept	The fundamental concept of this modelling is that it is based on baselining and demand shifting, i.e., possible shifts of demand blocks earlier than the corresponding baseline profile.
Activations	The baseline demand profile is broken down into separate activations, and it is investigated if it can be shifted earlier in time.
Shift always earlier	An activation can be shifted earlier only by a specific time window in order to ensure comfort; a later activation may result in comfort violation (e.g., insufficient hot water supply).
Mean Power	The optimization problem considers the mean power of an activation over its duration for possible demand shifting.
Formulation	The formulation is based on cost minimization. The main aim is to help demand response offering to local flexibility markets. If prices are available, it also helps shift demand to time intervals when price is lower.
Price	If a price profile is available, then it can be considered to shift demand when price is lower, otherwise there are possibly many alternatives for shifting demand.
Time between activations	There is a minimum time between consecutive activations.
Overlap between optimized and baseline profiles	A partial overlap is not allowed; a full overlap, i.e. no change - return baseline as is, is acceptable if no optimization solutions are available.

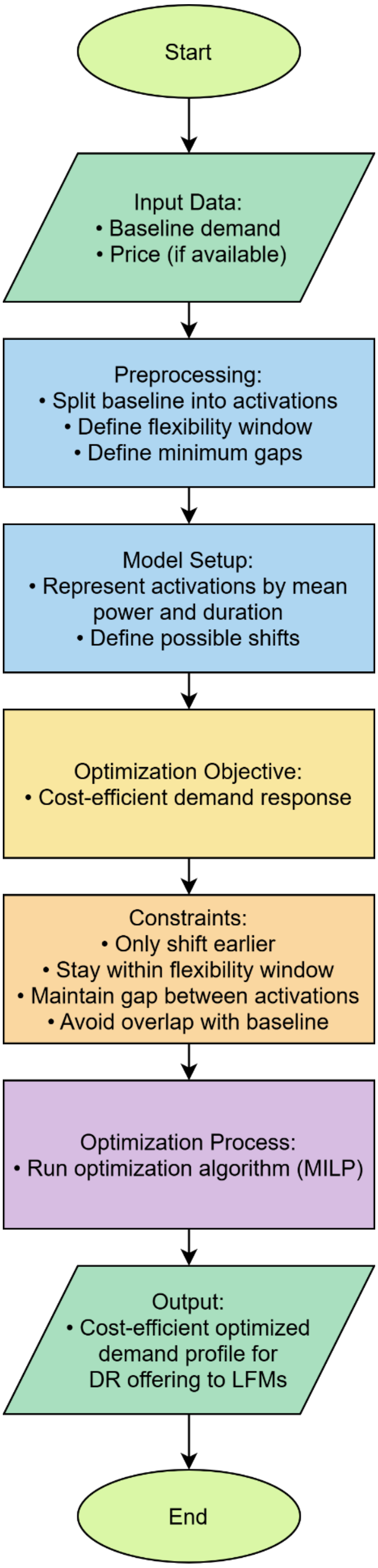


Figure 18: Flexibility Optimization Flowchart.

3.3.3 DR Initialization Service

The DR Initialization Service aims to deliver a user interface (UI) to facilitate the participation of occupants in flexibility activities. It provides a user-friendly way for them to provide input on the assets that will participate in the DR activities, as well as on individualized preferences or even constraints on the control of these assets. For instance, the occupants can define operational constraints, such as temperature set points for HVAC systems during the summer or winter period or a timeframe in which hot water must always be available. In this way, BFMS ensures that the flexibility actions align with the comfort preferences and daily routines of the occupants, without disrupting their behavior.

Its operation is ensured by a front-end and a back-end module (Figure 19). The Front-End (UI) Module is the user-facing part of the service and provides a graphical interface that allows the occupant to interact with BFMS. Through this UI, users can input their preferences and set boundaries for Demand Response participation. The Back-End Module consists of two sub-modules: the Prosumer/Asset Management Sub-module (managing the list of available flexibility assets within the user's building and linking them to the user) and the DR Settings & Preferences Persistence Sub-module (receiving, processing and storing the choices made by the user in the UI).

The occupant uses the Front-End (UI) to select which assets can participate in DR events and to define their comfort constraints. This information is sent to the Back-End, where it is managed and stored. Finally, these validated user settings are communicated to BFMS (core) system. BFMS then uses these preferences as constraints when calculating and dispatching flexibility, ensuring that all actions respect the occupant's comfort.

The use of this component is not strictly necessary for the end-to-end execution of OPENTUNITY's Demand Response (DR) schemes. However, its integration offers significant added value by incorporating end-users' preferences, ensuring a more tailored and user-centric approach to flexibility management. In this way, it reduces the need for constant manual adjustments, allowing occupants to participate in DR campaigns with confidence, knowing that their preferences will be respected.

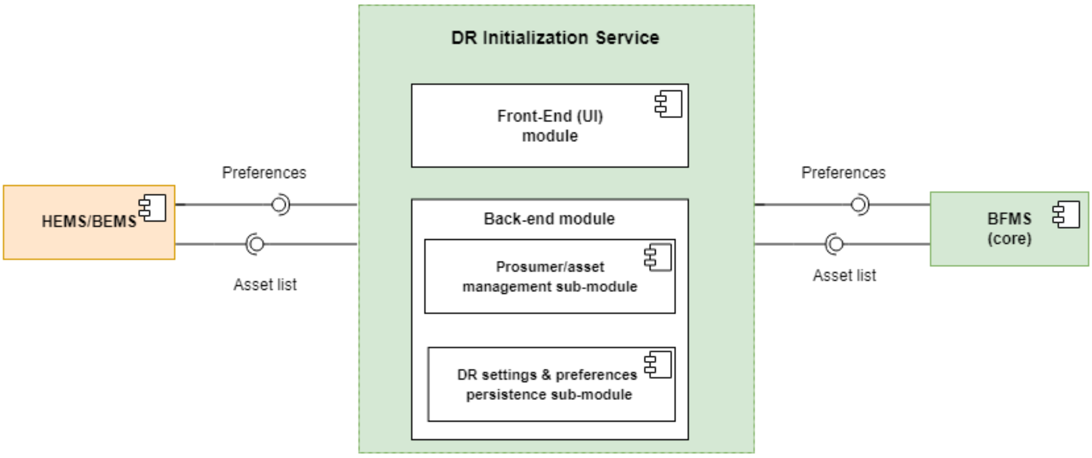


Figure 19: Component diagram of the DR Initialization Service

3.3.3.1 Usage walkthrough

This section serves as a brief demonstration for the DR Initialization Service, helping to understand the functionalities of the tool and ensuring that users will utilize it to provide their preferences and constraints. The following walkthrough illustrates the typical user journey based on screenshots of the service.

Upon access of the dedicated UI of the DR Initialization Service, users are shown the first page of the UI (Figure 20). The navigation panel on the left presents three main options: Asset Selection, Asset Preferences and General Settings. Each section guides users through a distinct phase of configuring their participation in the Demand Response scheme.

The process begins with the Asset Selection page (Figure 20), where users identify which devices within their household or building will participate in the DR program. Each listed asset is displayed alongside its device name, type and unique identifier (UUID). The user simply activates or deactivates participation by checking or unchecking the corresponding box in the Active column. For example, in the screenshot provided, the asset "Living Room AC" has been activated as part of the DR program, while the asset "Water Heater" remains inactive. This step allows the user to explicitly select the devices that will contribute to flexibility events, in this way offering a high degree of control and transparency. Once the desired devices are selected, the user finalizes their choices by clicking the Apply Changes button.

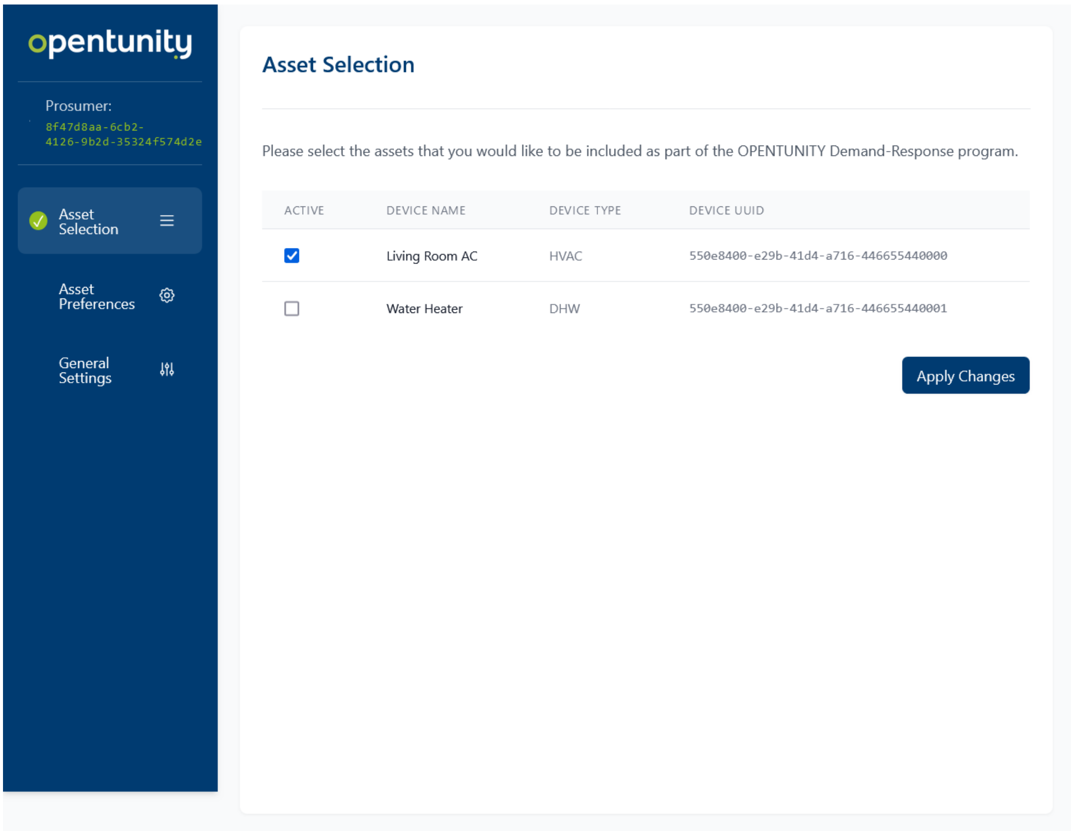


Figure 20: DR Initialization Service UI, Asset Selection

Following asset selection, the user navigates to the Asset Preferences section (Figure 21), where more granular settings can be defined for each active asset. The user can select the type of assets that they wish to see (for instance HVAC or DHW), on the upper right corner of the screen. Following this selection, the assets of the specific type will appear. For HVAC assets, the user may select the level of granted control, i.e., temperature set point and status (ON /OFF).

Below this, the user can provide their preference on the temperature range that they are comfortable with during the winter and the summer period. For DHW assets (Figure 22), the user can decide if they wish to declare a timeframe during which hot water should always be available. Once the desired choices are made, the user finalizes this section by clicking the Apply Changes button.

Lastly, the General Settings section (Figure 23) allows the user to review or modify global preferences. The date and time of the last changes is shown and the user is offered the option to reset all preferences to default values. Once again, should the user decide to reset their preferences, the Apply Changes button finalizes this section.

Each provided preference of the user is registered and used to update BFMS with the options and constraints selected by the user. Through this guided process, the DR Initialization Service changes a potentially technically complex task into an accessible and transparent user experience. In this way, the user participating in DR campaigns maintains a decisional power and is offered a human-centered flexibility engagement.

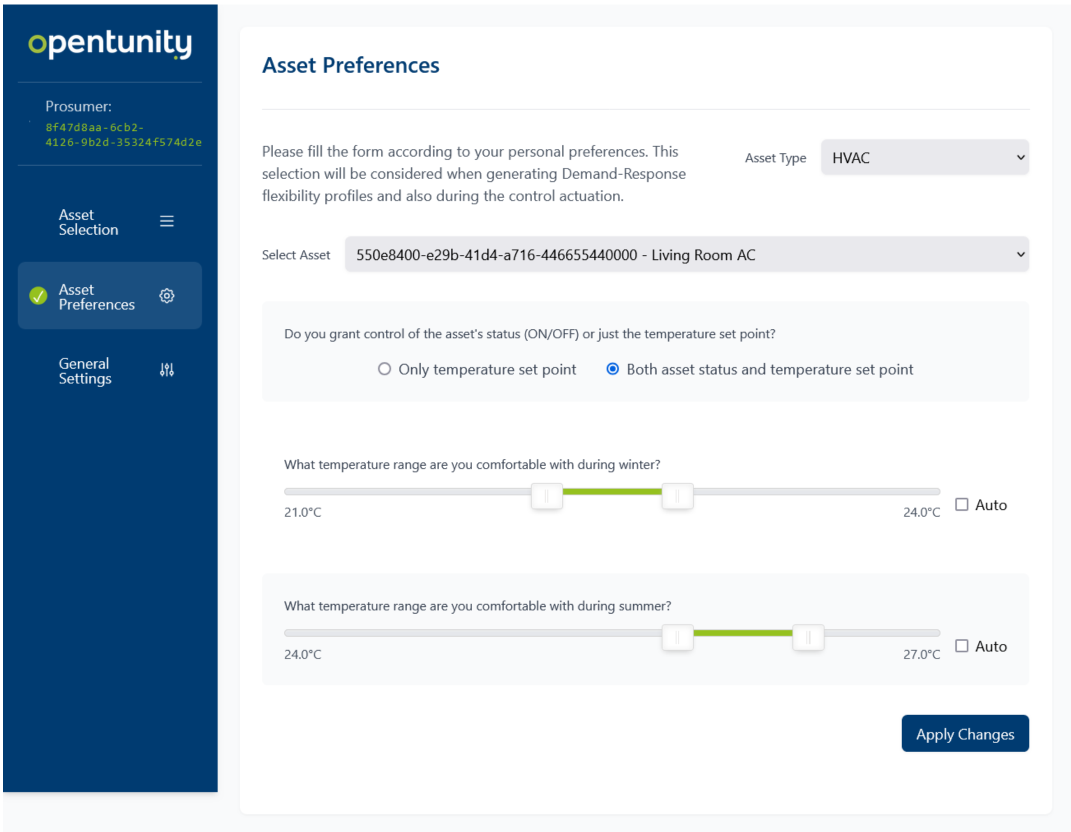


Figure 21: DR Initialization Service UI, HVAC Asset Preferences

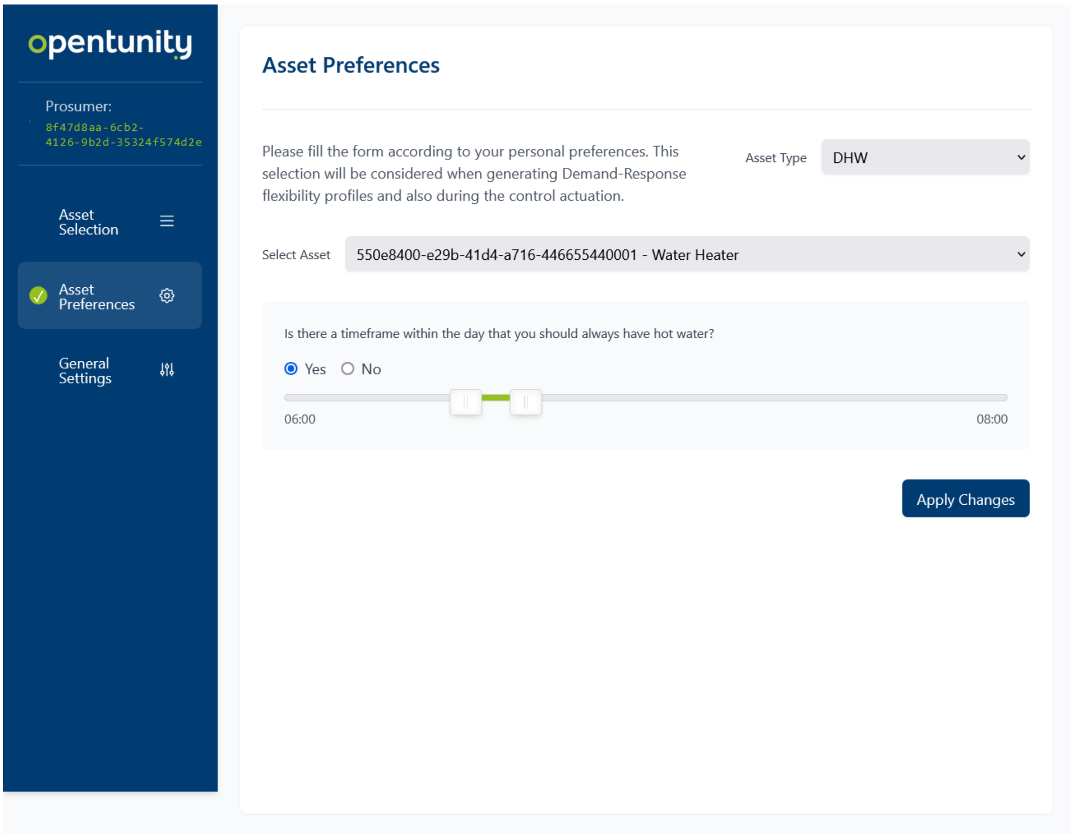


Figure 22: Initialization Service UI, DHW Asset Preferences

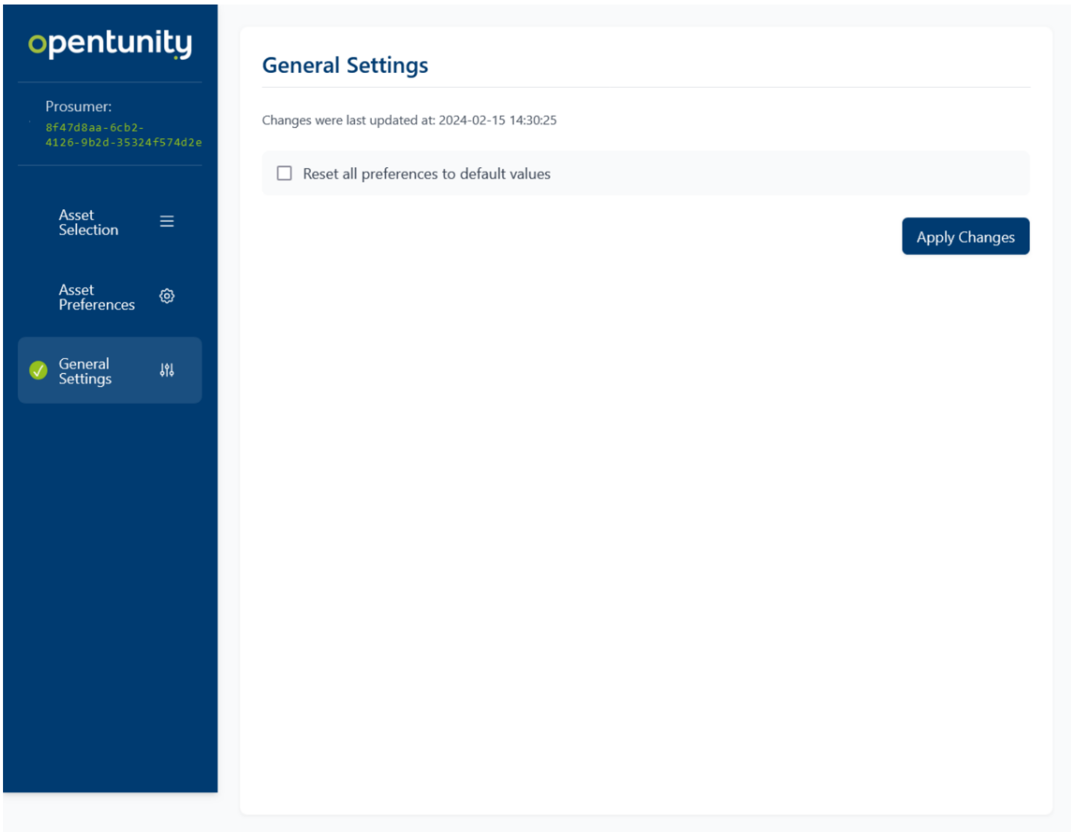


Figure 23: Initialization Service UI, General Settings

3.3.4 Control Dispatching Service

The Control Dispatching Service is the part of BFMS that interacts with the different HEMS/BEMS of the project to enable control of the individual building assets. Upon acceptance of a flexibility bid in the NODES platform, this service will evaluate the feasibility of forecasted flexibility actions and will send control signals to the selected pilot assets. The system prioritizes non-intrusive operation, ensuring comfort levels are preserved while executing flexibility or load-shifting scenarios.

During operation, the module integrates direct flexibility dispatch requests with flexibility forecasts (e.g., power consumption curves). For example, HVAC operation may be adjusted by modifying thermostat setpoints while making sure that comfort is maintained. EWHs may be switched on or off based on flexibility needs. The Service takes into account the current status of an asset. For instance, If the device is already in the requested mode/status, a command is not dispatched (e.g., if asked status is ON and the device is currently ON, no command will be dispatched). Additionally, if a DR signal violates comfort boundaries, it will not be dispatched (e.g., only change the temperature set point for HVAC but not turn ON/OFF nor change operation mode). Also, in the case of EWH, as per the pilot users request, a pre-heating sequence might take place to ensure minimal intrusion to their daily routine.

Furthermore, the system offers the possibility of a manual override by the user (e.g., setting the controlling relay to manual ON or OFF or adjusting the HVAC setpoint), in this way allowing the occupants to increase their comfort level if needed. Once the DR event concludes, subsequent control actions return devices to their baseline operation.

Depending on the HEMS/BEMS configuration, the system supports bi-directional communication with the building's infrastructure, enabling real-time control and continuous monitoring of asset performance.

The Control Dispatching Service consists of the Control Design Module and the Control Dispatcher Module (Figure 24). The former receives the accepted flexibility request and disaggregated information from the FSP.

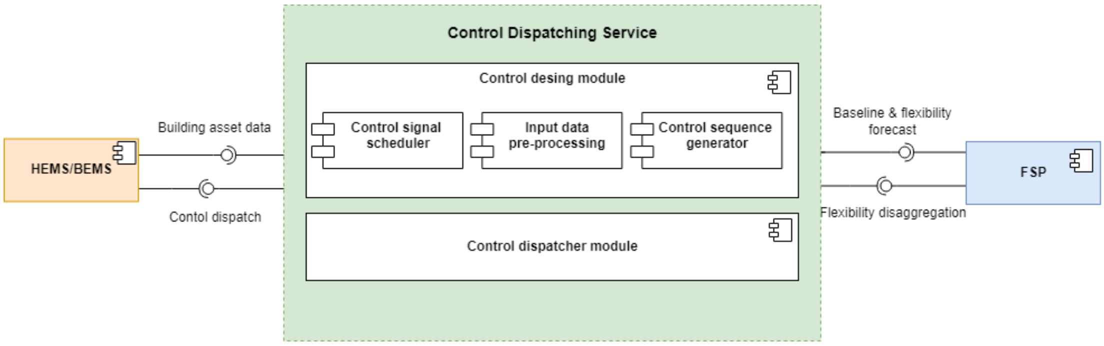


Figure 24: Control Dispatching Service component diagram

Through the Input Data Pre-processing submodule, it gathers and prepares data, such as the baseline and optimized energy consumption, as well as user preferences. It then proceeds with the Control Sequence Generator, a sub-module that designs the asset specific command sequences needed to achieve the required power adjustment. Lastly, the Control Signal Scheduler determines the timing for executing the generated control sequences to ensure the flexibility market participation.

The Control Dispatcher Module uses the scheduled control sequences from the Design Module, performs just in-time check of the asset availability and dispatches the command signals to the appropriate IoT controllers and actuators within the building.

3.3.5 Flexibility Service Provider

The FSP serves as the bridge between BFMS and the NODES platform, by aggregating the asset specific energy consumption acquired from the former to large-scale flexibility values to the latter. It collects day-ahead baseline and flexibility forecasts from multiple assets and buildings, processes this data and creates a single, aggregated flexibility offer that is submitted to the NODES platform for bidding. Once a flexibility bid is accepted on NODES, the accepted bid returns to the FSP that breaks down the total power adjustment requirement into specific disaggregated values for each participating asset. In this way, the FSP functions as a two-way service, converting numerous small-scale flexibility into a market bid and then transforming a market purchase back into asset specific actions.

The FSP is composed of two sub-modules that manage the flow of flexibility to and from the energy market: the Flexibility Aggregator and the Flexibility Disaggregator (Figure 25). The Flexibility Aggregator module is responsible for creating a market-ready flexibility product. Its operation begins with the Input Data Pre-processing sub-module, which receives and standardizes day-ahead Baseline and Flexibility Forecasts from multiple assets. This data is then passed to the Aggregate Flexibility Generator sub-module, whose function is to sum the flexibility potential from all buildings, creating a single and larger flexibility offer. Finally, the Output Data Post-processing sub-module formats this aggregated offer according to market specifications and sends it to the NODES platform for bidding.

Conversely, the Flexibility Disaggregator module handles the breakdown of a cleared flexibility bid. It includes an Input Data Pre-processing sub-module, which receives an accepted bid from the NODES platform. This bid, which specifies the total amount of aggregate flexibility required and the timing, is then provided into the Disaggregate Flexibility Generator sub-module. This sub-module performs the task of disaggregating the total requirement and allocating the necessary adjustments back to the individual assets. The Output Data Post-processing sub-module then formats these asset specific adjustments and sends them to BFMS for execution.

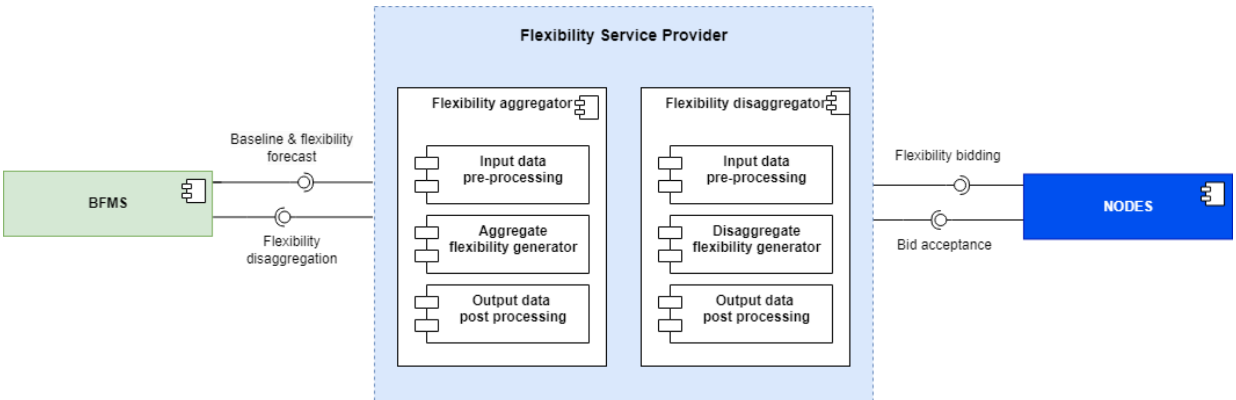


Figure 25: Flexibility Service Provider component diagram

3.3.6Sequence diagrams

This section provides detailed sequence diagrams showcasing the operation of BFMS, in alignment with the relevant Use Cases as defined in Deliverable 2.1 Technical foundations. Specifically, for UC 1.8 HEMS/BEMS DR optimization and local flexibility management and UC 1.9 Initialization of HEMS/BEMS Demand Response strategy.

Starting with the forecast of baseline and flexibility energy consumption values, Figure 26 illustrates the interactions between the different modules of BFMS and steps required for the calculation of business as usual and flexibility operation. HEMS/BEMS provide the required building data for different modules of BFMS, i.e., the Asset Profiling, the Comfort Profiling and the Building Thermal Modeling modules. Each of these modules then proceeds with the modeling of the asset, comfort and building elements of the specific investigated case, as described in section 3.3.2 Profiling and Flexibility Forecasting Orchestration Service (core service). The outputs from all three preceding modules are then fed to the Flexibility Forecasting Module. This module uses all inputs to calculate the building's day-ahead baseline energy consumption and its available flexibility potential.

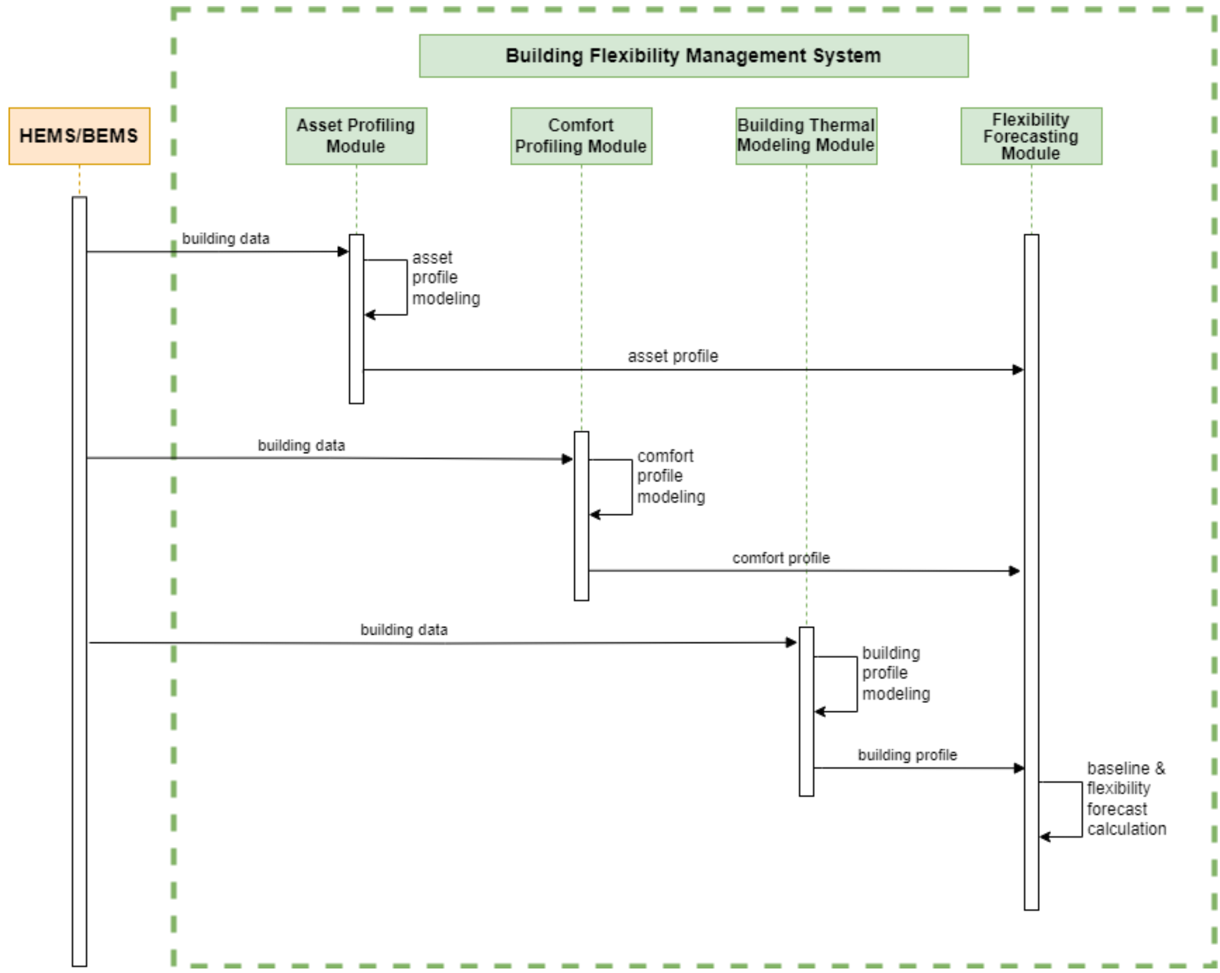


Figure 26: Baseline and flexibility forecast sequence diagram

Figure 27 presents the interaction flow between different services and tools to capture and implement user preferences in relation to their participation in the flexibility market. The process is initiated by the occupant through the dedicated DR Initialization Service UI, as described in section 3.3.3 DR Initialization Service. DR Initialization Service will request the full asset list from the

HEMS/BEMS and associate them with the specific user. The user will then select which assets can participate in DR events and under what conditions (e.g., temperature setpoints, timeframe, etc.). This information is then sent from the DR Initialization Service to the other modules of BFMS for storage and use by subsequent processes.

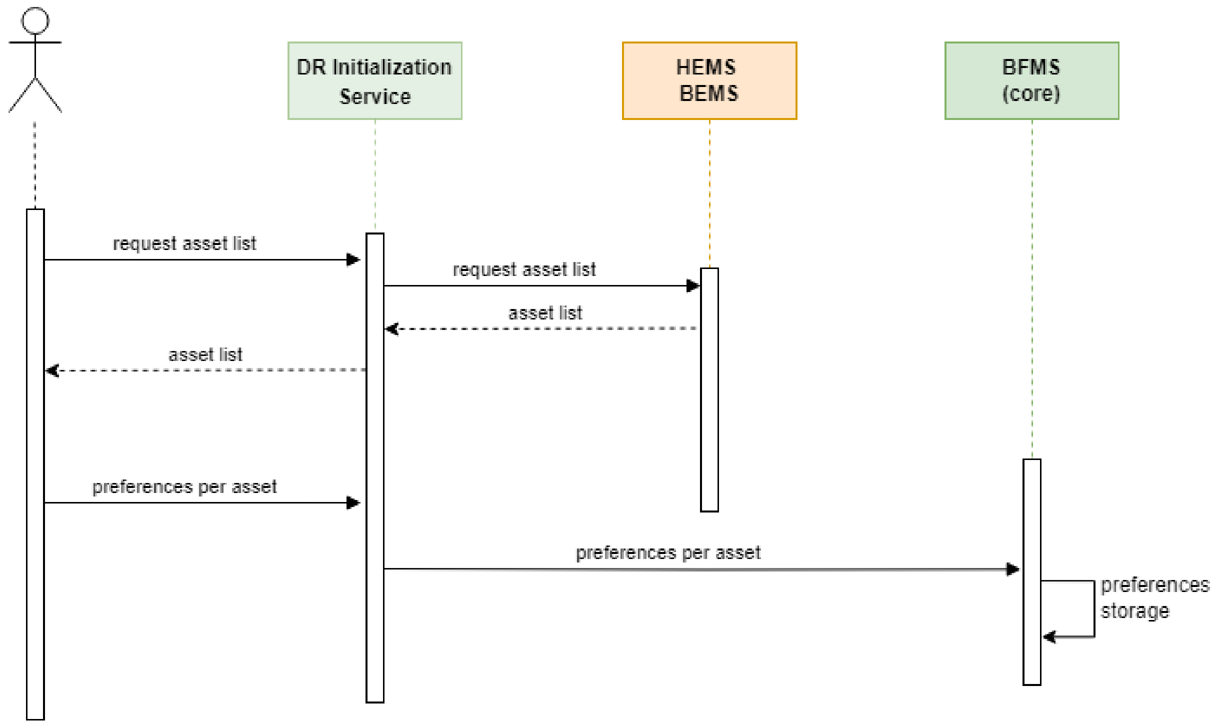


Figure 27: DR initialization service sequence diagram

Lastly, Figure 28 illustrates the sequence of processes for the translation of an accepted flexibility bid into control actions at the pilot sites. The Flexibility Forecasting Module provides the baseline and flexibility forecasts to the FSP. The FSP will firstly aggregate the available flexibility (as described in section 3.3.5 Flexibility Service Provider) and proceed with the flexibility bid to the NODES platform. There, a bid negotiation and possibly acceptance process will take place, also depending on the offers of the SOs for the period in question. The accepted bid returns to the FSP which will at this point proceed with the disaggregation of the accepted flexibility to the controllable assets. The FSP communicates the disaggregated desired flexibility to the Control Design Module that will investigate the feasibility of the DR flexibility execution.

In case of a positive outcome, the Control Design Module will then schedule the asset specific command actions needed to achieve the required power adjustment. It will communicate this information to the Control Dispatcher Module, that will perform a control backlog check and will issue near real-time command signals to the participating assets. A reverting control signal is later sent to return the asset to its previous state.

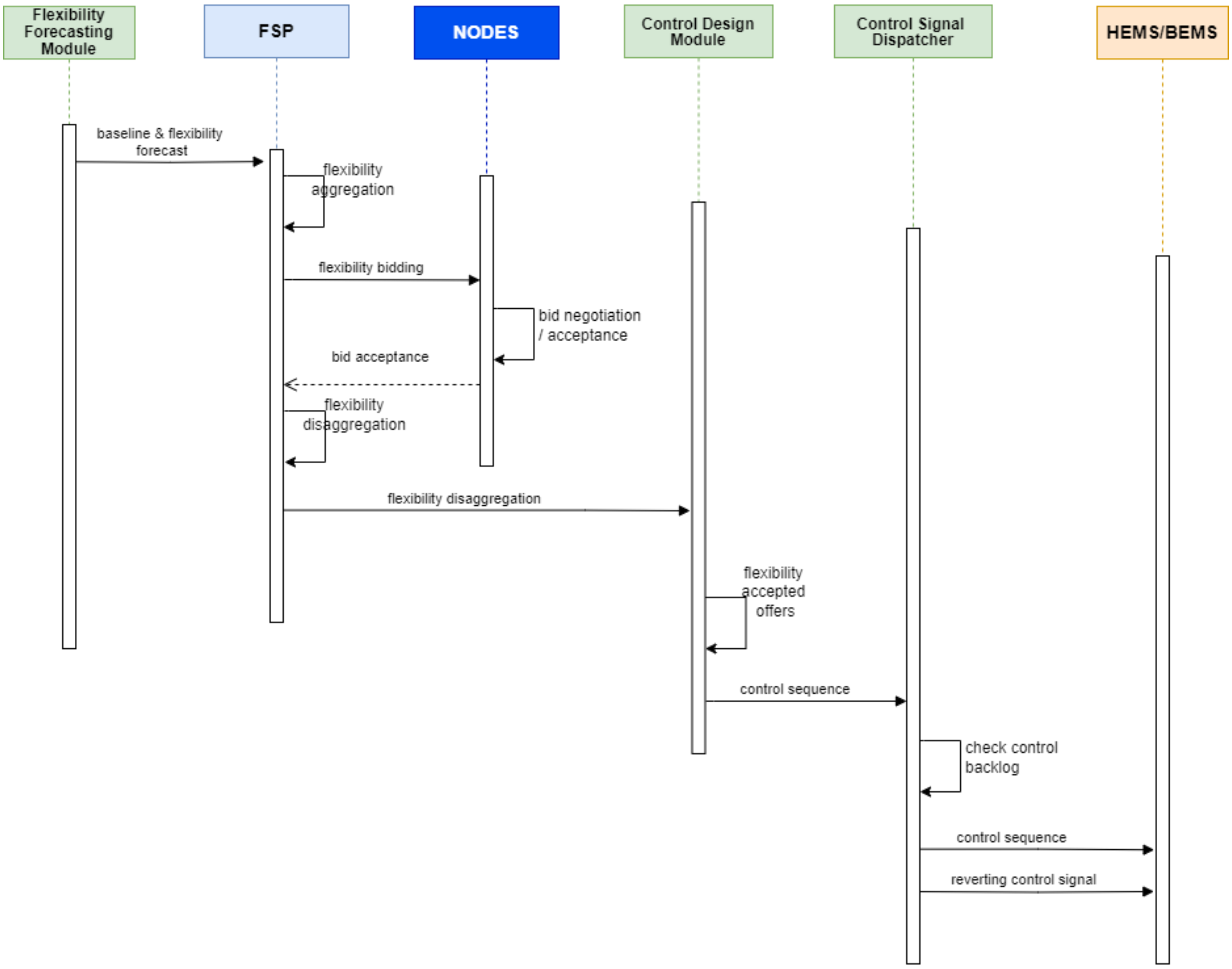


Figure 28: Control dispatch sequence diagram

3.3.7 Integration with other ecosystems

The operation of BFMS depends on its seamless and continuous integration with the broader OPENTUNITY ecosystem. This concerns various tools and services, such as the Data Space, the HEMS/BEMS platforms at the different pilot sites of the project and the NODES platform, that allow the retrieval of necessary information from the flexibility assets, the dispatch of control commands and the retrieval of BFMS's forecasted values. This section presents the requirements and the implementation of the integration process with these tools.

3.3.7.1 Data Space

The integration of BFMS with the Data Space ensures secure, reliable and interoperable data exchange among participating entities, both within OPENTUNITY ecosystem and potentially outside of it, namely interested external entities. The Data Space approach allows BFMS to connect with other platforms without altering its existing business logic or data models; data remains securely at its source, while self-sovereign data exchange principles ensure that each partner retains full control and ownership of its data. This integration occurs through a multiple steps approach, including the participant's registration to the Identity Management System, the account approval by the Data Space administrator, the submission of basic information, the reception of an API KEY and the deployment of a virtual machine.

A more elaborate description of the onboarding and integration of the BFMS and the HEMS/BEMS to the data space is presented in detail in Deliverable 3.2 Federated Data Exchange Infrastructure Architecture (v2).

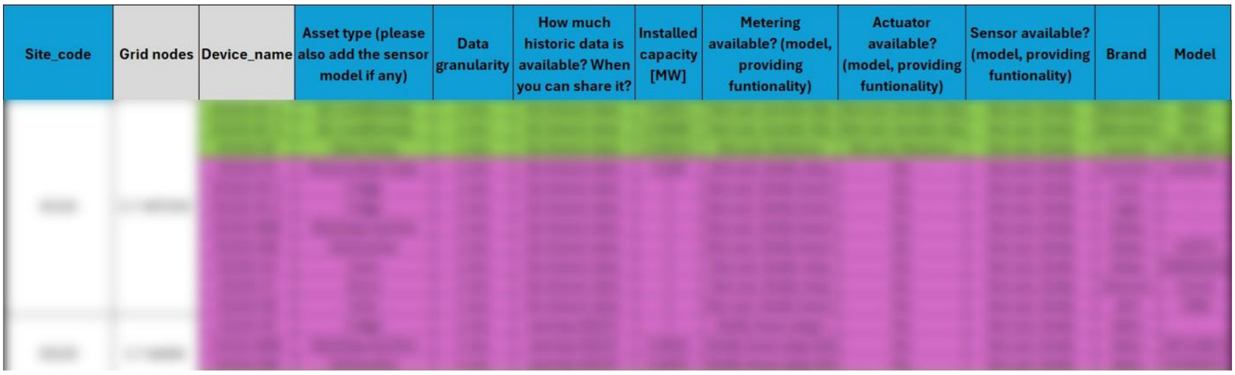
3.3.7.2 HEMS/BEMS

BFMS interacts with three different HEMS/BEMS at three distinct pilot sites, requiring tailored integration approaches to accommodate their specific architecture and capabilities. It must be able to receive timeseries data for power consumption, indoor temperature and asset status, as well as to send control commands for assets participating in the flexibility market.

The inclusion of assets was performed through a structured approach. First, an asset registry was created for each pilot site to record and organize all available flexibility assets. This registry included details such as unique identifiers, asset types, capacity, the presence of metering, actuator, and sensor devices, along with the level of data granularity and availability.

Second, the registry was circulated among the pilot partners, who were responsible for reviewing and populating it with accurate and complete data relevant to their respective sites. This collaborative effort ensured consistency and completeness across all pilot implementations. Figure 29 illustrates one of the pilot registries, with sensitive information blurred for confidentiality purposes.

Third, adaptations to the data models were carried out to harmonize the structure and content of the registries, facilitating their integration with other system components and ensuring interoperability across different pilots. In addition, individual wrappers were developed to accommodate the specificities of each of the HEMS and BEMS deployed across the pilot sites. These wrappers enabled the seamless translation of heterogeneous data formats and communication protocols into a unified structure, ensuring reliable interaction between the various building management systems and the overarching platform.



Site_code	Grid nodes	Device_name	Asset type (please also add the sensor model if any)	Data granularity	How much historic data is available? When you can share it?	Installed capacity [MW]	Metering available? (model, providing funtionality)	Actuator available? (model, providing funtionality)	Sensor available? (model, providing funtionality)	Brand	Model

Figure 29: Asset registry per pilot

3.3.7.2.1. Greek Pilot

The integration of BFMS with the assets at the Greek pilot site supports both data retrieval from household and building devices and remote control of controllable loads. The integration is implemented through a set of secure RESTful APIs, providing standardized interfaces for both data exchange and actuation over the Data Space.

Real-time and historical data retrieval occurs via the following endpoint:

```
https://projects.energylabs-ht.eu/api/data/{equipmentUUID}?startDate={start}&endDate={end}&interval=60&intervalUnit=minute
```

Each monitored device is uniquely identified by a UUID, the endpoint returning time-series data in JSON format, including the value (the measured or averaged value, e.g., temperature, power) and the startDate / endDate (the time interval of each measurement).

The following is an example of a response:


```json
{
  "_embedded": {
    "TimeseriesData": [
      {
        "value": "19.467",
        "startDate": "2025-01-31T21:00:00.000Z",
        "endDate": "2025-01-31T21:59:59.999Z"
      },
      {
        "value": "19.1",
        "startDate": "2025-01-31T22:00:00.000Z",
        "endDate": "2025-01-31T22:59:59.999Z"
      }
    ]
  },
  "_links": {
    "self": {
      "href": "https://projects.energylabs-ht.eu/api/data/5ff96c13-4862-11ec-9d97-960000500b98?endDate=2025-01-31T23:00:00.000Z&fillGaps=true&dataGroupingMethod=average&startDate=2025-01-31T21:00:00.000Z&interval=60&intervalUnit=minutes&page=0&size=100"
    },
    "timeseries-info": {
      "href": "https://projects.energylabs-ht.eu/api/items/5ff96c13-4862-11ec-9d97-960000500b98"
    }
  },
  "page": {
    "size": 100,
    "totalElements": 2,
    "totalPages": 1,
    "number": 0
  }
}
```


The control of assets is also handled via a dedicated control API, enabling the dispatch of actuation commands (e.g., ON/OFF, setpoint temperature adjustment) to controllable devices.

Control requests are executed using a POST method at the following endpoint:

```
curl --location 'https://projects.energylabs-ht.eu/api/control' \
--header 'Content-Type: application/json' \
--data '{
   "itemUUID":"01703609-f1fc-11ec-9da0-960000500b98",
   "loadType":"hvacControl"
   "value":"OFF"
}'
```

For both fetch and control actions the respective UUIDs are required. In order to get access to these UUIDs the following method is used.

Request:

```
curl --location 'https://projects.energylabs-ht.eu/api/prosumers/bc7f2662-f1fb-11ec-9da0-960000500b98' \
--header 'Authorization: Bearer ******
```

Response:


```
{
   "prosumer-uuid": "bc7f2662-f1fb-11ec-9da0-960000500b98",
   "prosumer-label": "OPT055",
   "prosumer-type": "R",
   "prosumer-country": "GR",
   "prosumer-global-loads": [
      {

         "load-uuid": "01703608-f1fc-11ec-9da0-960000500b98",
         "load-type": "TOTAL METERING",
         "load-label": "Main Meter",
         "load-items": [

                        {

            "item-uuid": "0170360a-f1fc-11ec-9da0-960000500b98",
            "item-metric-type": "Energy",
            "item-unit": "kWh",
            "item-controllability": false,
            "_links": {
               "timeseries-data": {
                  "href": "https://projects.energylabs-ht.eu/api/data/0170360a-f1fc-11ec-9da0-960000500b98"
               },
               "timeseries-info": {
                  "href": "https://projects.energylabs-ht.eu/api/items/0170360a-f1fc-11ec-9da0-960000500b98"
               }
            }
         }........
       ]
    }]
}
```

3.3.7.2.2. Spanish Pilot

The interaction between BFMS and the assets of the Spanish pilot is enabled through dedicated API endpoints developed and maintained by ETRA. These interfaces facilitate the bidirectional exchange of information, including both the retrieval of device monitoring data and the execution of control commands over the available assets. The communication follows a RESTful architecture, using secure HTTPS connections and JSON-formatted request bodies over the Data Space.

The data retrieval interface provides access to the time-series measurements, as collected from the devices of the pilot. The endpoint for querying this information is:

```
https://opentunity-metrics.k8s.etra-id.com/gateway
```

The retrieved data includes real-time and historical measurements such as active power, temperature and operational status. The request body follows a JSON structure and includes several parameters defining the query scope, filters, aggregation level and temporal grouping.

In parallel with the data retrieval interface, a second API is available to support the remote control of the available assets. The control endpoint is accessible via the following URL:

```
https://opentunitysensibo-control.k8s.etra-id.com/docs#/
```

This endpoint exposes a Swagger-based REST API, providing a structured description of available control operations and their corresponding parameters. Through this interface, control signals are sent to devices, such as switching devices on or off, modifying the operational mode or adjusting the setpoint temperatures.

3.3.7.2.3. Swiss Pilot

The integration of BFMS with the assets from the Swiss pilot site ensures the necessary data retrieval from the controllable devices. This integration is realized through the AEM R&D API, which provides access to a range of measurement points from devices and sensors, allowing BFMS to gather real-time and historical operational data. To obtain access, partners must submit a request to AEM explaining the purpose of the data use, including the relevant Work Package or Task and justifying the need for real-time information. BFMS retrieves these data through secure API calls over HTTPS, using authentication credentials and an anonymization procedure for data protection compliance. The control of the assets of the Swiss pilot is performed by SUPSI and is described in section 2.2 The data managed by ELIOT is structured hierarchically in four defined levels:

- **Realm or pilot,** a high-level grouping of different installations.
- **Site,** typically represents a building or location where several devices are installed.
- **Zone,** a sub-division of a site (physical or logical) where a subset of devices is located.
- **Device,** a wide range of elements that provide measurements and other information from the field. Devices can be located inside a zone or directly from the site.

In parallel, ELIOT has also been integrated with residential air conditioning devices to enable demand response actions. For this purpose, a dedicated API has been developed, allowing the flexibility service provider to remotely interact with the HVAC units installed in households. This integration has been achieved through the Sensibo API, which enables the modification of setpoint temperatures of

AC units connected to a Sensibo smart AC controller, and supports the activation of flexibility offers. Through this new development, ELIOT can both display real-time information and actively send control commands.

AEM R&D platform - . Due to confidentiality constraints, detailed procedures and configurations related to API access cannot be disclosed in this deliverable.

3.3.7.3 BFMS endpoint

The forecasted baseline and flexibility values of BFMS at consumer and asset levels can be retrieved through a dedicated endpoint. This is described in detail in Section 3.6 Interfaces & API documentation.

3.3.7.4 NODES platform

The case of the NODES flexibility market platform requires an automated or manual process for bid submission and reception of market results. NODES requires aggregated flexibility offers from the FSP, which are generated based on the in-turn forecasts of the BFMS. Therefore, the integration with NODES is managed by the FSP, which acts as the market participant, while the role of BFMS is to supply the FSP with the necessary data to formulate these offers.

To facilitate this interaction, an adapter has been developed to interface directly with the NODES platform. This component serves as a middleware layer that automates data exchange between the BFMS and NODES, ensuring the timely submission of bids, retrieval of market outcomes and general alignment with the platform's communication protocols. The adapter manages authentication, request formatting and response handling, in this way minimizing manual intervention and enhancing the reliability and speed of the overall process.

The NODES platform ensures secure and standardized communication with external systems through a well-defined API framework. A specific authentication process must be followed to ensure a secure connection. The API is fully documented and offers a flexible design that allows authorized users to query data efficiently using standard GET requests. Due to confidentiality constraints, detailed procedures and configurations related to API access cannot be disclosed in this deliverable.

3.4 Technology stack and Implementation tools

BFMS component utilizes state-of-the-art tools and technologies to deliver a modular, efficient, highly configurable and replicable solution; most of the technologies used are free and/or open source. The BFMS suite is deployed as microservices based on containerization technologies (Docker), offering increased flexibility in deployment options/platforms, horizontal scaling capabilities and enhanced observability features. A concise summary of the individual libraries and technologies used is presented in Table 2 below.

Table 2: Libraries and Technologies used in BFMS

Library/Technology Name	Version	License	Module/service used
Python	3.12	Free, Open Source PSF License (Python Software Foundation License)	DR Initialization Service, Profiling & Flexibility Orchestration Service, Control Dispatching Service
Python main libraries used: numpy, scipy pandas, scikit-learn, prefect, docker, pydantic, pika, apscheduler, orjson, httpx, python-cron	(latest versions at the time of writing)	Free, Open Source Various open-source licenses (e.g., BSD, MIT, Apache 2.0)	
Java	21 LTS	Free, Open Source GPL v2 with Classpath Exception	Profiling Storage and Persistence Service, HEMS/BEMS Integration Layer
Java main libraries used: spring MVC, hibernate	(latest versions at the time of writing)	Free, Open Source Various open-source licenses (e.g., Apache 2.0, LGPL v2.1)	
PostgreSQL	17.x	Free, Open Source PostgreSQL License (Permissive, similar to BSD)	Profiling Storage and Persistence Service
HTML, CSS, Javascript TailwindCSS		Free, MIT License	DR Initialization Service Front-End UI
Redis	7.4	BSD 3-Clause License	Control Dispatching Service
Docker (with docker compose)	28.0, 2.33	Apache License 2.0	Overall BFMS deployment
Nginx	1.27	BSD 2-Clause License	Profiling Storage and Persistence Service
Tomcat	8.5	Apache License 2.0	Profiling Storage and Persistence Service

3.5 Input/Output parameters

The following table outlines the input and output requirements for the successful operation of BFMS component.

Table 3: Input and output parameters of BFMS component

Comfort profiling	
Input parameters	
Space air temperature	Indoor temperature collected from building sensors (˚C)
Outdoor air temperature	Outdoor temperature sourced from weather data (˚C)
Occupancy data	Optional data collected from the occupancy profiling component
Output parameters	
Baseline thermal comfort boundaries for each space	Occupants' comfort zone (lower and upper boundaries) during occupied periods
Flexibility thermal comfort boundaries for each space	Extended comfort zone (lower and upper boundaries) enabling flexibility during occupied periods
Building thermal modelling	
Input parameters	
T_i	Indoor temperature in ˚C (timeseries, received from OPENTUNITY HEMS/BEMS)
T_{ext}	Outdoor temperature in ˚C (timeseries)
Q_s	Global horizontal irradiance in W/m^2 (timeseries)
Q_g	Power consumption of other building electric loads
$Q_{h1, \ldots n}$	HVAC power consumption (n = number of HVAC systems)
Output parameters	
Trained RC model for each space	Matrices estimated during system identification
AC Split Units with Inverters	
Input parameters	
x_1	Boolean for device status (0 = off)
x_2	Thermostat setpoint temperature in ˚C (timeseries)
x_3	Hourly outdoor temperature in ˚C (timeseries)
x_4	Hourly global horizontal irradiance in W/m^2 (calculated from cloud coverage and weather data)
x_5	Hourly outdoor relative humidity in % (timeseries)

x_6	Indoor temperature in ˚C (timeseries)
f(x)	Device power consumption in W (timeseries)
Output parameters	
GP(x)	Trained GP model for forecasting HVAC energy consumption
Air-to-Air Constant Volume Heat Pumps and Electric Heaters	
Input parameters	
k	Number of clusters, determined during preprocessing
Power consumption (x_1)	Time-series data from metering devices
Operational status (x_2)	Boolean indicator (0 for off, 1 for on) from monitoring devices
Output parameters	
Power clusters	Different levels of device power consumption, used for optimized control
Air-to-Water and Water-to-Water Heat Pumps	
Input parameters	
N_k	Cluster number assigned by k-means for the timestamp
T_m	Minute of the hour (e.g., 0, 15, 30, 45)
TH	Hour of the day [0, 23]
T_D	Day of the week [0, 6]
Frequent kWs	Aggregated consumption data by minute, hour and day
External temperature	Outdoor temperature from local weather stations
Seasonality	Periodic time-series feature from seasonal decomposition
Output parameters	
Forecasted Power Profile (P_1, P_2, … P_n)	Day-ahead predicted power consumption (time series) for the heat pumps
EWH Model	
Input parameters	
k	Number of clusters for k-means clustering
x_1	Device power consumption in W (timeseries)
x_2	Boolean for device status (0 = off)

Output parameters	
$P_{1,2,\ldots n}$	Forecasted EWH power profile for n timesteps in W (timeseries)
Flexibility Forecasting Service	
Input	
Baseline thermal comfort boundaries for each space	Occupants' comfort zone (lower and upper boundaries) during occupied periods
Flexibility thermal comfort boundaries for each space	Extended comfort zone (lower and upper boundaries) enabling flexibility during occupied periods
Trained RC model for each space	Matrices estimated during system identification
Asset Profiles	Baseline power day-ahead profile in W (timeseries)
Consumer Preferences (from DR Initialization)	User defined comfort preferences
Output	
Baseline Forecasted Profile	Baseline power day-ahead profile in W (timeseries)
Flexibility Forecasted Profile	Upward and downward power day-ahead profile in W (timeseries)
DR Initialization Service	
Input	
Asset list	List of available assets per user
Consumer Preferences (from user)	User defined comfort preferences
Output	
Consumer Preferences	User defined comfort preferences
Control Dispatching Service	
Input	
Flexibility disaggregation values (from FSP)	Upward and downward power day-ahead profile in W (timeseries) per asset
Asset list	List of available assets per user
Output	
Control actions	**Control actions per asset**

Flexibility Service Provider	
Input	
Baseline Forecasted Profile	Baseline power day-ahead profile in W (timeseries)
Flexibility Forecasted Profile	Upward and downward power day-ahead profile in W (timeseries)
Bid acceptance (from NODES)	Accepted flexibility bid
Output	
Flexibility bid	**Aggregated flexibility bid**
Flexibility disaggregation	**Disaggregated flexibility values**

3.6 Interfaces & API documentation

Results of BFMS are made available to other tools and services of the OPENTUNITY ecosystem through dedicated endpoints. Table 4 and Table 5 present a detailed description of the endpoints for the Prosumer-level baseline and flexibility profiles and the Asset-level baseline and flexibility profiles respectively. The two tables provide a description of the endpoint, the HTTP method and important parameters. In addition, an example sample response is presented, detailing the data structure and expected format. The response includes detailed time-stamped data for both baseline and flexibility forecasts, indicating expected energy use and the potential for flexibility during each time interval.

Table 4: Prosumer-level baseline and flexibility profiles

Endpoint	**/prosumers/flexibility/forecast/{prosumerUUID}**
Description	An endpoint for requesting and receiving the aggregated baseline and the flexibility profiles (timeseries) of a **Prosumer** under a specific optimisation scenario for a given timeframe and a given sample rate
HTTP Method	**GET**
Parameters	**`prosumerUuid`**: string A unique identifier for a Prosumer
	`startDate`: string The start date of the timeframe in `YYYY-MM-DDTHH:mm:ss.SSSZ` format
	`endDate`: string The end date of the timeframe in `YYYY-MM-DDTHH:mm:ss.SSSZ` format
	`period`: string The profiles (timeseries) sample rate in minutes *Default value:* `15 minutes`
	`requestType`: string The requested optimization/flexibility scenario:

	Value to be used: `explicitDemandResponse`
	`intraDay`: boolean A parameter for requesting the re-evaluated (intraday forecast) of the baseline and flexibility profiles if applicable and available *Value to be used*: `false`
	`baselineEstimationMethod:` string The preferred method for querying the baseline profile; P = Power and E = Energy *Value to be used*: `absolute`
	`flexibilityEstimationMethod`: string The preferred method for querying the flexibility profile; P = Power and E = Energy *Value to be used*: `absoluteP`
Sample Response (redacted)	(see below)


```
{
    "requestInfo": {
        "uuid": "bf2d076a-202a-4798-b0ad-3da2ea7f6467",
        "flexibilityEstimationMethod": "absoluteP",
        "requestType": "explicitDemandResponse",
        "requestScope": "prosumerLevel",
        "prosumerUuid": "afe61bab-0eab-11ee-9da7-961000500b98",
        "startDate": "2025-10-10T00:00:00.000+0000",
        "endDate": "2025-10-11T00:00:00.000+0000",
        "createdAt": "2025-10-20T12:15:50.443+0000",
        "period": "15 minutes",
        "intraDay": false
    },
    "baselineForecasting": [
        {
            "dtStart": "2025-10-10T00:00:00.000+0000",
            "dtEnd": "2025-10-10T00:15:00.000+0000",
            "mean": 0.0,
            "unit": "W",
            "createdAt": "2025-10-09T04:25:20.736+0000"
        },
        {
            "dtStart": "2025-10-10T00:15:00.000+0000",
            "dtEnd": "2025-10-10T00:30:00.000+0000",
            "mean": 0.0,
            "unit": "W",
            "createdAt": "2025-10-09T04:25:20.736+0000"
        },

…

        {
            "dtStart": "2025-10-10T23:30:00.000+0000",
            "dtEnd": "2025-10-10T23:45:00.000+0000",
            "mean": 0.0,
            "unit": "W",
            "createdAt": "2025-10-09T04:25:20.736+0000"
        },
        {
            "dtStart": "2025-10-10T23:45:00.000+0000",
            "dtEnd": "2025-10-11T00:00:00.000+0000",
            "mean": 0.0,
            "unit": "W",
```

```
                "createdAt": "2025-10-09T04:25:20.736+0000"
            }
        ],
        "flexibilityForecasting": [
            {
                "dtStart": "2025-10-10T00:00:00.000+0000",
                "dtEnd": "2025-10-10T00:15:00.000+0000",
                "meanUp": 0.0,
                "meanDown": 0.0,
                "unit": "W",
                "createdAt": "2025-10-09T04:25:20.736+0000"
            },
            {
                "dtStart": "2025-10-10T00:15:00.000+0000",
                "dtEnd": "2025-10-10T00:30:00.000+0000",
                "meanUp": 0.0,
                "meanDown": 0.0,
                "unit": "W",
                "createdAt": "2025-10-09T04:25:20.736+0000"
            },

…

            {
                "dtStart": "2025-10-10T23:30:00.000+0000",
                "dtEnd": "2025-10-10T23:45:00.000+0000",
                "meanUp": 0.0,
                "meanDown": 0.0,
                "unit": "W",
                "createdAt": "2025-10-09T04:25:20.736+0000"
            },
            {
                "dtStart": "2025-10-10T23:45:00.000+0000",
                "dtEnd": "2025-10-11T00:00:00.000+0000",
                "meanUp": 0.0,
                "meanDown": 0.0,
                "unit": "W",
                "createdAt": "2025-10-09T04:25:20.736+0000"
            }
        ]
}
```


Table 5: Asset-level baseline and flexibility profiles

Endpoint	**/assets/flexibility/forecast/{assetUUID}**
Description	An endpoint for requesting and receiving the aggregated baseline and the flexibility profiles (timeseries) of an **Asset** under a specific optimization scenario for a given timeframe and a sample rate.
HTTP Method	**GET**
Parameters	**`assetUUID`**: string A unique identifier for a Prosumer
	`startDate`: string The start date of the timeframe in `YYYY-MM-DDTHH:mm:ss.SSSZ` format
	`endDate`: string The end date of the timeframe in `YYYY-MM-DDTHH:mm:ss.SSSZ` format

	period: string The profiles (timeseries) sample rate in minutes *Default value:* `15 minutes`
	requestType: string The requested optimization/flexibility scenario: *Value to be used*: `explicitDemandResponse`
	intraDay: boolean A parameter for requesting the re-evaluated (intraday forecast) of the baseline and flexibility profiles if applicable and available *Value to be used*: `false`
	baselineEstimationMethod: string The preferred method for querying the baseline profile; P = Power and E = Energy *Value to be used*: `absolute`
	flexibilityEstimationMethod: string The preferred method for querying the flexibility profile; P = Power and E = Energy *Value to be used*: `absolute`
Sample Response (redacted)	(see below)


```
{
    "requestInfo": {
        "uuid": "586bfb25-0f34-45a1-aa9f-cb40029224ef",
        "flexibilityEstimationMethod": "absoluteE",
        "requestType": "explicitDemandResponse",
        "requestScope": "assetLevel",
        "assetUuid": "12ad26ba-1eac-11ee-9da7-961000500b98",
        "startDate": "2025-10-10T00:00:00.000+0000",
        "endDate": "2025-10-11T00:00:00.000+0000",
        "createdAt": "2025-10-20T12:57:12.671+0000",
        "period": "15 minutes",
        "intraDay": false
    },
    "baselineForecasting": [
        {
            "dtStart": "2025-10-10T00:00:00.000+0000",
            "dtEnd": "2025-10-10T00:15:00.000+0000",
            "mean": 0.0,
            "unit": "W",
            "createdAt": "2025-10-09T04:25:20.736+0000"
        },
        {
            "dtStart": "2025-10-10T00:15:00.000+0000",
            "dtEnd": "2025-10-10T00:30:00.000+0000",
            "mean": 0.0,
            "unit": "W",
            "createdAt": "2025-10-09T04:25:20.736+0000"
        },

…

        {
            "dtStart": "2025-10-10T23:30:00.000+0000",
            "dtEnd": "2025-10-10T23:45:00.000+0000",
```

```
                "mean": 0.0,
                "unit": "W",
                "createdAt": "2025-10-09T04:25:20.736+0000"
            },
            {
                "dtStart": "2025-10-10T23:45:00.000+0000",
                "dtEnd": "2025-10-11T00:00:00.000+0000",
                "mean": 0.0,
                "unit": "W",
                "createdAt": "2025-10-09T04:25:20.736+0000"
            }
        ],
        "flexibilityForecasting": [
            {
                "dtStart": "2025-10-10T00:00:00.000+0000",
                "dtEnd": "2025-10-10T00:15:00.000+0000",
                "meanUp": 0.0,
                "meanDown": 0.0,
                "unit": "W",
                "createdAt": "2025-10-09T04:25:20.736+0000"
            },
            {
                "dtStart": "2025-10-10T00:15:00.000+0000",
                "dtEnd": "2025-10-10T00:30:00.000+0000",
                "meanUp": 0.0,
                "meanDown": 0.0,
                "unit": "W",
                "createdAt": "2025-10-09T04:25:20.736+0000"
            },

…

            {
                "dtStart": "2025-10-10T23:30:00.000+0000",
                "dtEnd": "2025-10-10T23:45:00.000+0000",
                "meanUp": 0.0,
                "meanDown": 0.0,
                "unit": "W",
                "createdAt": "2025-10-09T04:25:20.736+0000"
            },
            {
                "dtStart": "2025-10-10T23:45:00.000+0000",
                "dtEnd": "2025-10-11T00:00:00.000+0000",
                "meanUp": 0.0,
                "meanDown": 0.0,
                "unit": "W",
                "createdAt": "2025-10-09T04:25:20.736+0000"
            }
        ]
    }
```


3.7 Pre-validation activities & Preliminary tests

3.7.1 Theoretical Pre-Validation Case Studies

This section shows the results of the updated model previously presented in Section 3.3.2.4. The flexibility forecasting algorithm initially identifies the number of demand assets and their corresponding type. Then for each asset, the number of activations is obtained, together with start time and end time for each one, and the mean power for each activation interval. Note that if there is only one 15-minute time interval between two consecutive activations, they are combined into a single one.

The case study considers three demand assets (two DHW devices and an HVAC). For the given input, N = 3 (number of demand assets), a = [2, 2, 3] (activations per demand asset), and T = 96 (number of time steps). Figure 30, Figure 31, and Figure 32 illustrate the baseline power consumption of DHW 1, DHW 2, and HVAC, respectively.

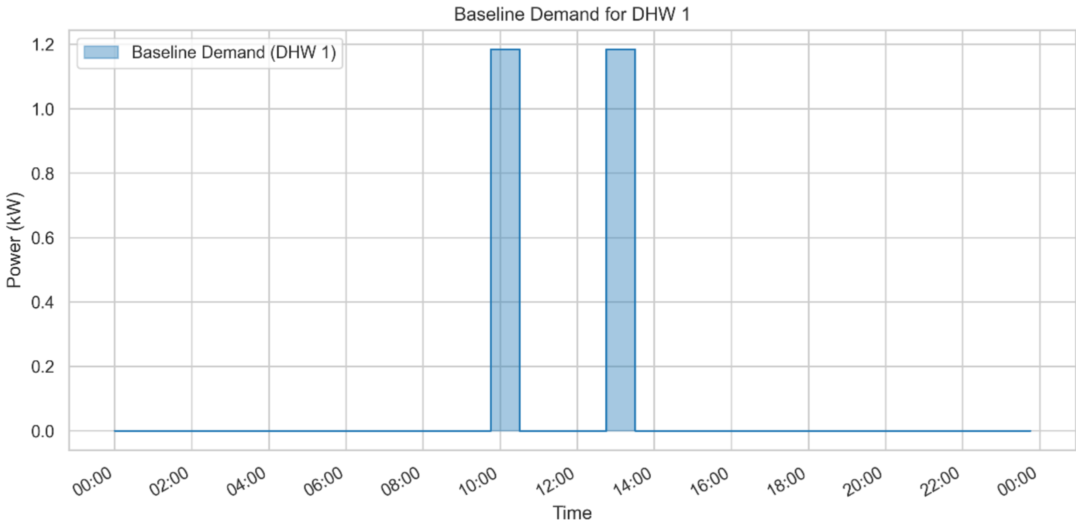


Figure 30: Power consumption of DHW 1 – Baseline

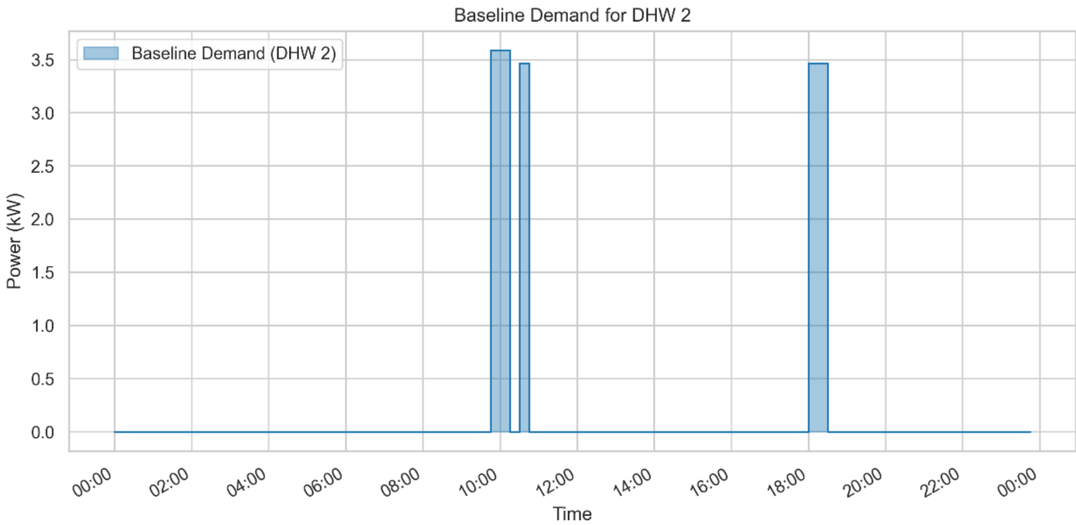


Figure 31: Power consumption of DHW 2 – Baseline

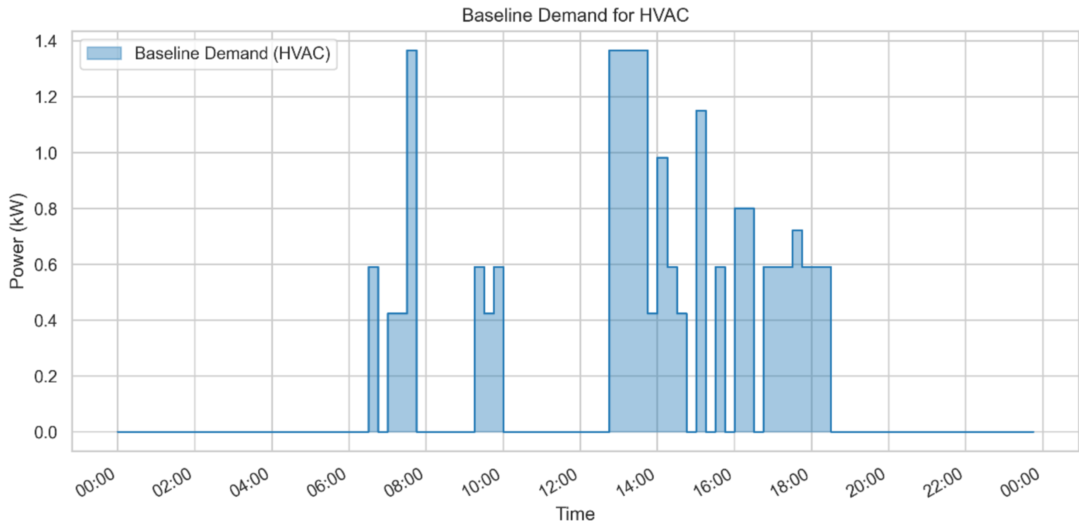


Figure 32: Power consumption of HVAC – Baseline

In the following subsections the results of two scenarios are presented, the first one with a price profile and a second one without prices.

3.7.1.1 Scenario with Price Profile

This section presents the results of the updated model given the existence of a price profile. The figures present each demand asset separately in order to make results more comprehensible.

Figure 33 shows optimized and baseline DHW 2 demand along with a price profile. There are two activations, the first one at around 10:00 and the second one at around 18:00. In both cases, activations are shifted earlier and there are no overlaps with the baseline profile. Moreover, they are temporally located within time intervals where price is the lowest (80 €/MWh).

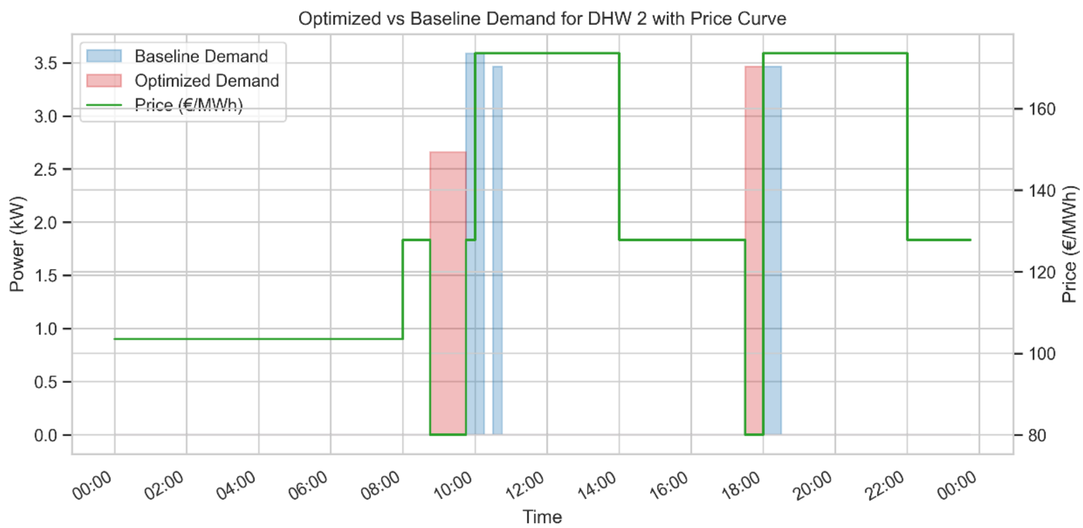


Figure 33: Optimized DHW 2 demand, baseline DHW 2 demand, and price for the updated model.

The optimized and baseline HVAC demand along with the price are shown in Figure 34. There are three activations in total. The first two (optimized activations) have no overlap with the baseline profile.

The third one fully overlaps with the baseline, because the duration of the activation is longer than the flexibility window, i.e., if it was shifted according to the flexibility window, there would still be overlap with the baseline profile, and this is something that the updated model does not allow.

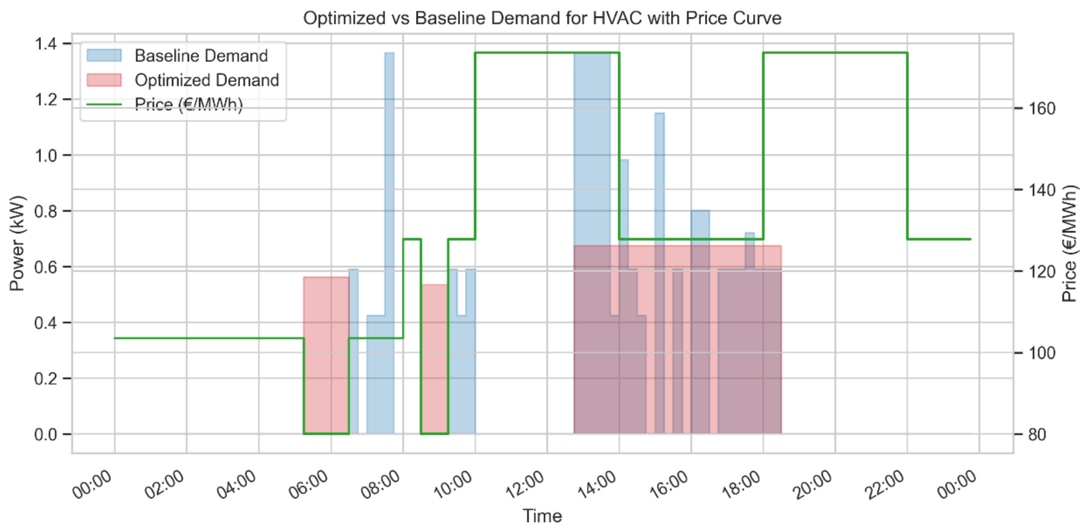


Figure 34: Optimized HVAC demand, baseline HVAC demand, and price for the updated model.

DHW 1 demand (optimized and baseline) along with the price profile are shown in Figure 35. There are two activations. The second one has no overlap with the baseline, and there is no lower price interval that can be shifted to. The first action has no overlap with the baseline as well, however, it could have been within a lower price time interval, but because there is a minimum time between consecutive activations (equal to three hours for DHW), the first activation has been shifted further earlier, at which electricity is priced at above 120 €/MWh (for part of that time interval).

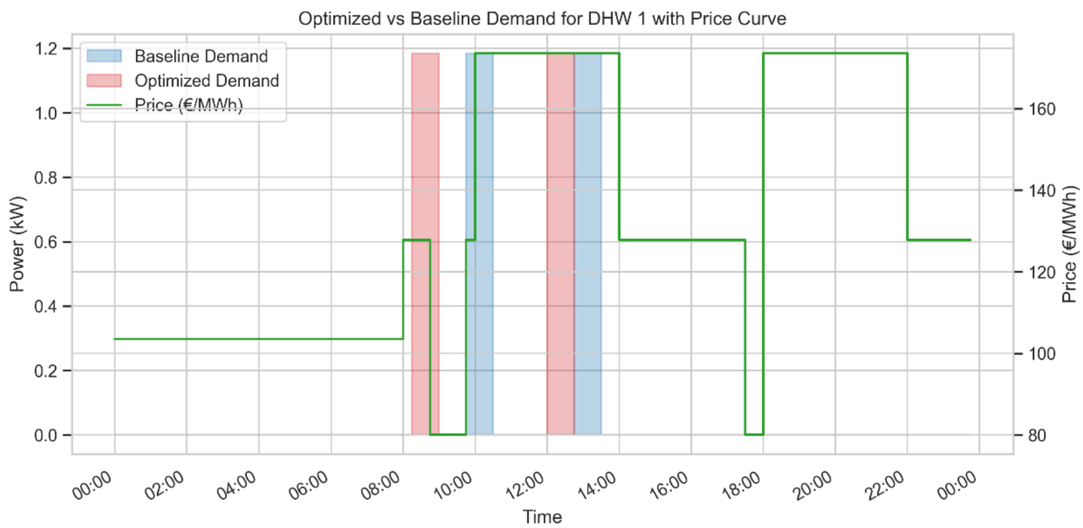


Figure 35: Optimized DHW 1 demand, baseline DHW 1 demand, and price for the updated model.

This section finally presents a specific test case to demonstrate that priority is given to demand response (DR) over cost minimization. The second activation of DHW 2 depicted in Figure 36 is shifted earlier to a time interval where price is higher compared to the baseline time interval. This shows that the main aim is to provide DR, while also being as cost-effective as possible.

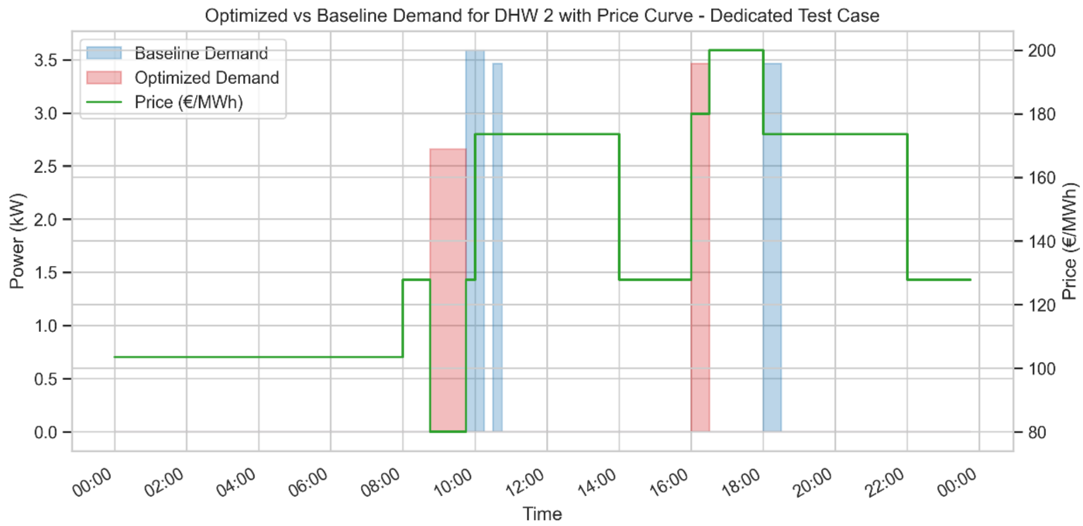


Figure 36: Dedicated test case to demonstrate the priority given to DR over cost minimization (second activation).

3.7.1.2 Scenario without Price Profile

This section shows the results of the updated model without a price profile. Since the updated model is based on a cost minimization formulation, in order to capture the absence of a price profile, the same value for the price (100 €/MWh) has been considered for all time steps. The results are presented in Figure 37 for DHW 1 only (similar results for DHW 2 and HVAC). It can be understood that the model now allows greater freedom in load shifting. This means that if no other constraints enforce otherwise, an activation can now have multiple realizations (i.e., multiple time intervals that can be shifted to) with the same objective function value.

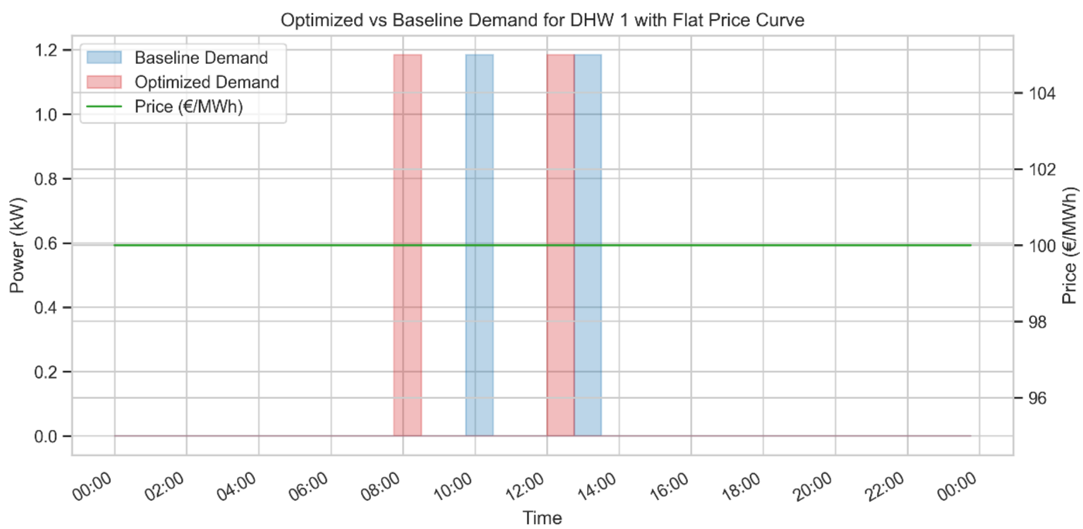


Figure 37: Optimized and baseline demand for DHW 1 only and no price (modelled as a flat price profile).

3.7.2 Actual end-to-end test results

This section presents actual test cases but also examines the impact of user preferences (related to T4.2) can have on the optimal solution when they are considered into the optimization problem. The new baseline (one DHW - Figure 38, one HVAC - Figure 39) is initially presented and then the results of the following scenarios are shown: 1) demand shifting not allowed by the user, 2) demand shifting allowed by the user and further preferences have not been introduced, 3) demand shifting allowed by the user and further preferences have been introduced. The above scenarios are considered with and without the impact of prices.

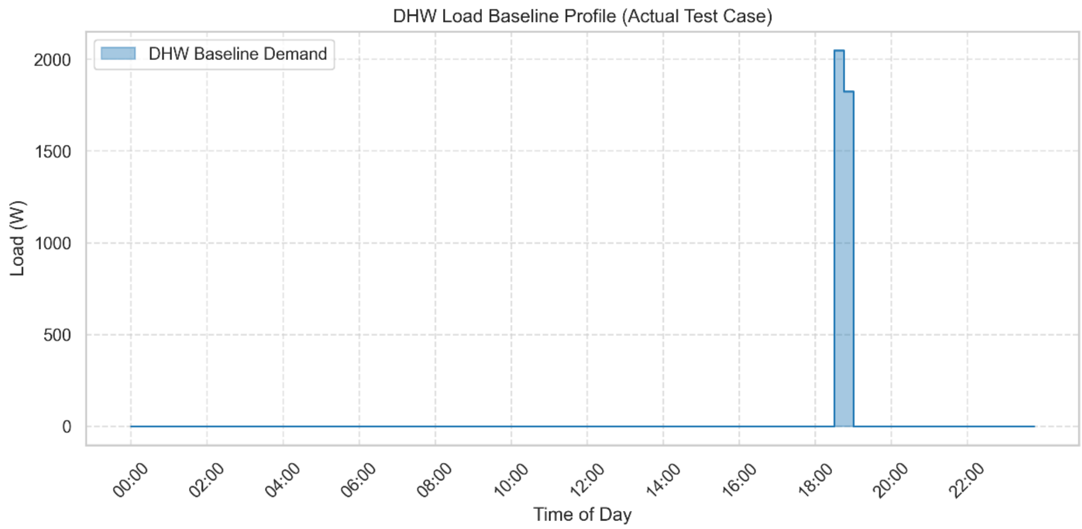


Figure 38: DHW Baseline Profile.

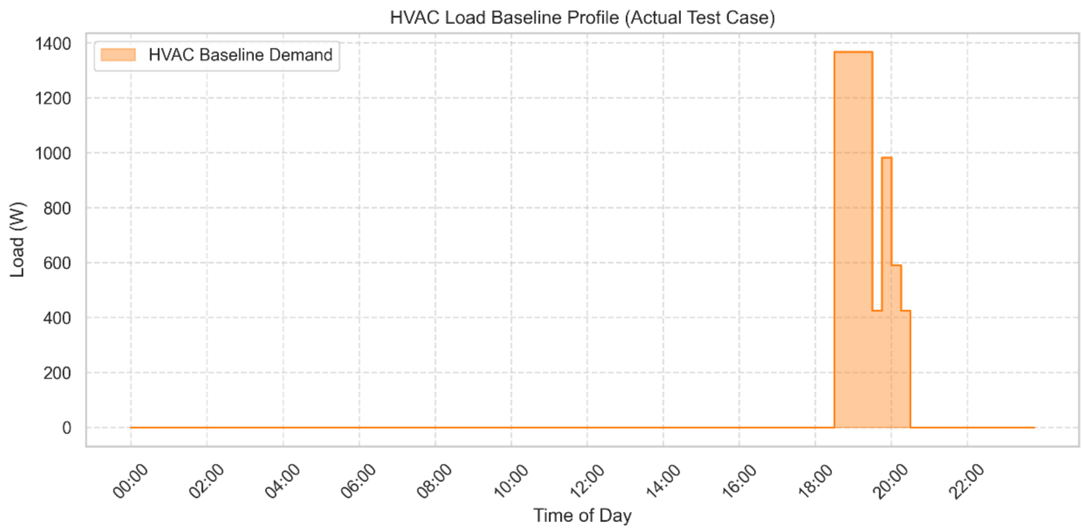


Figure 39: HVAC Baseline Profile.

3.7.2.1Scenarios with Price Profile

3.7.2.1.1. Demand shifting not allowed by the user

Figure 40 and Figure 41 show that the optimized load profile fully overlaps with the baseline as demand shifting is not allowed by the user. Note that the optimized profile is shown here in terms of mean power over the corresponding activation duration, as this is the way the demand profile is taken into account within the optimization problem.

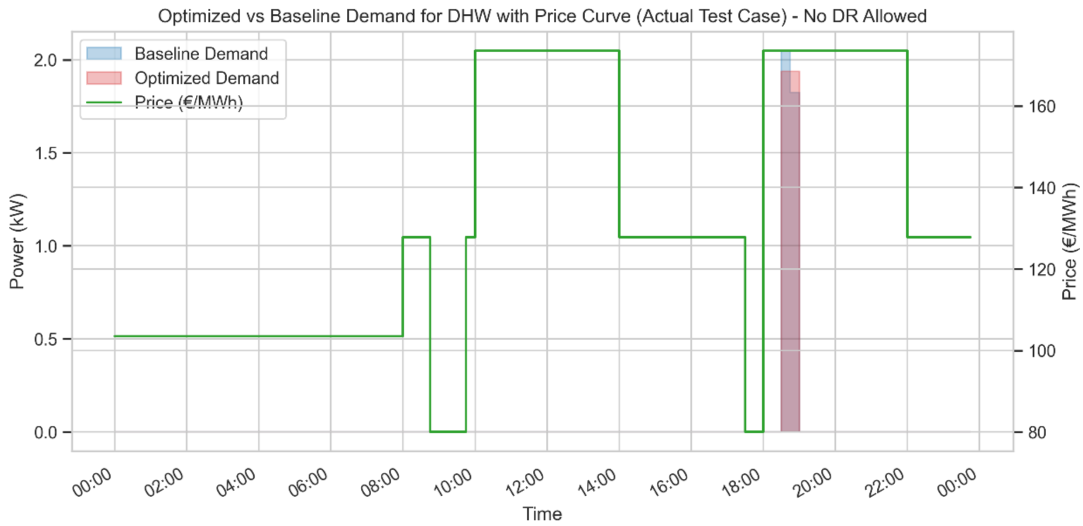


Figure 40: DHW Optimized Profile – Demand shifting not allowed by the user.

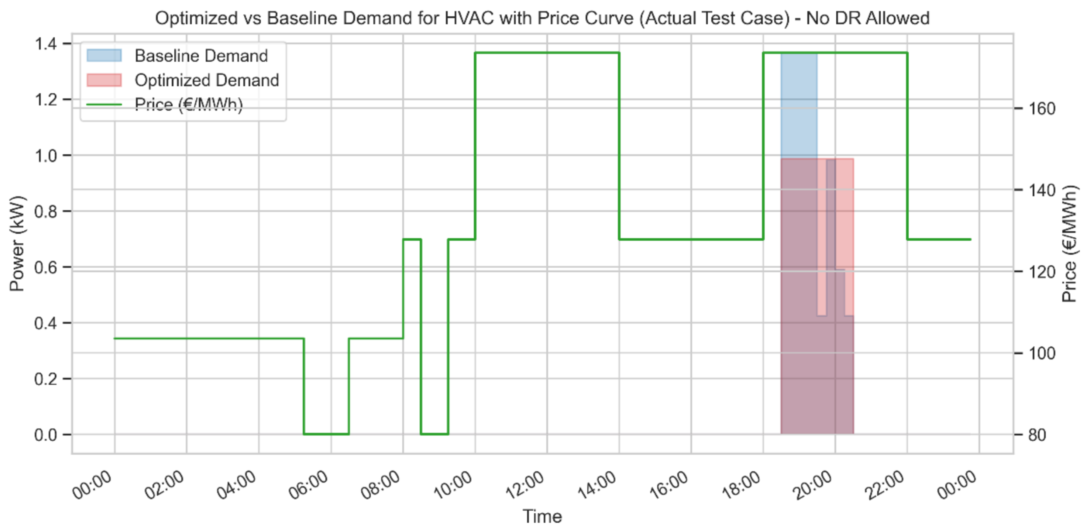


Figure 41: HVAC Optimized Profile – Demand shifting not allowed by the user.

3.7.2.1.2. Demand shifting allowed by the user and further preferences have not been introduced

Figure 42 demonstrates that the model delivers cost-efficient DR as DHW demand is shifted earlier when price is minimum. Figure 43 shows that HVAC demand is shifted two hours earlier, because the duration of the activation is equal to the flexibility window. If the duration was longer, shifting would not be allowed by the constraints of the optimization problem (as was the case with the HVAC demand profile in the theoretical pre-validation case studies section).

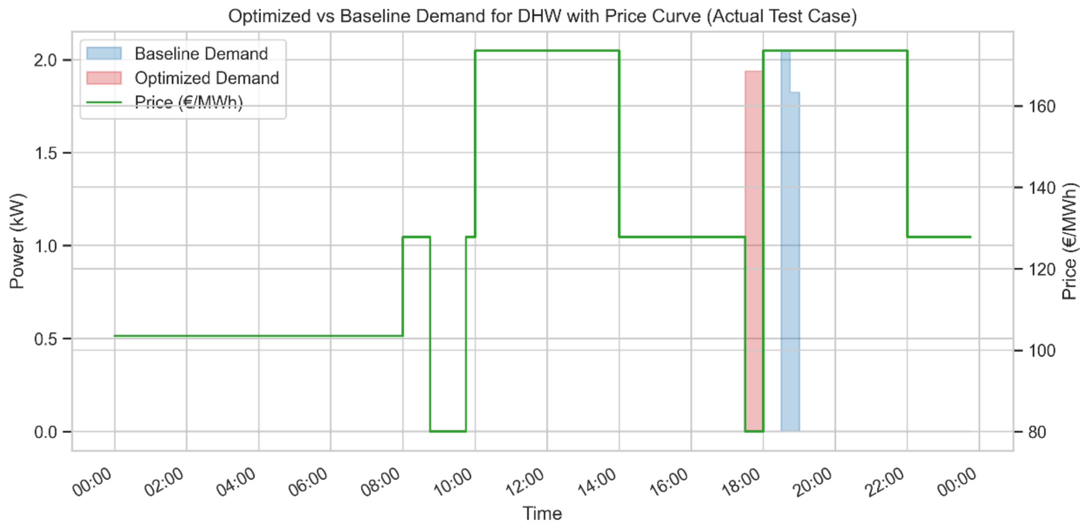


Figure 42: DHW Optimized Profile – Demand shifting allowed by the user and further preferences have not been introduced.

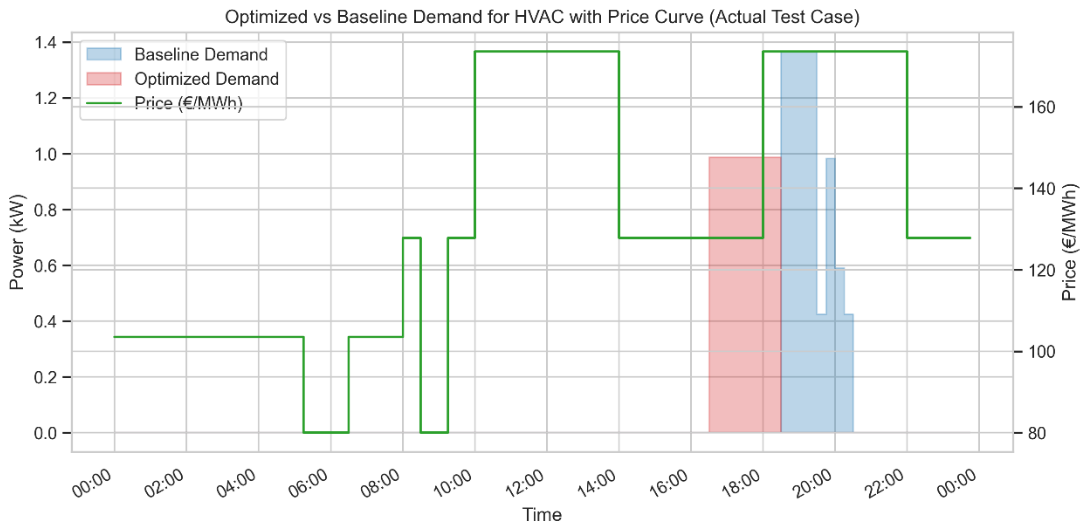


Figure 43: HVAC Optimized Profile – Demand shifting allowed by the user and further preferences have not been introduced.

3.7.2.1.3. Demand shifting allowed by the user and further preferences have been introduced

Assumption: 17:00 – 18:00 time interval is not allowed by the user.

The DHW single activation has been shifted at the earliest possible time interval (16:30 – 17:00) in order to satisfy the constraint that has been set by the user but also minimize the objective function. The user constraint could also allow an activation shift to 18:00 – 18:30, but this would correspond to a higher electricity price.

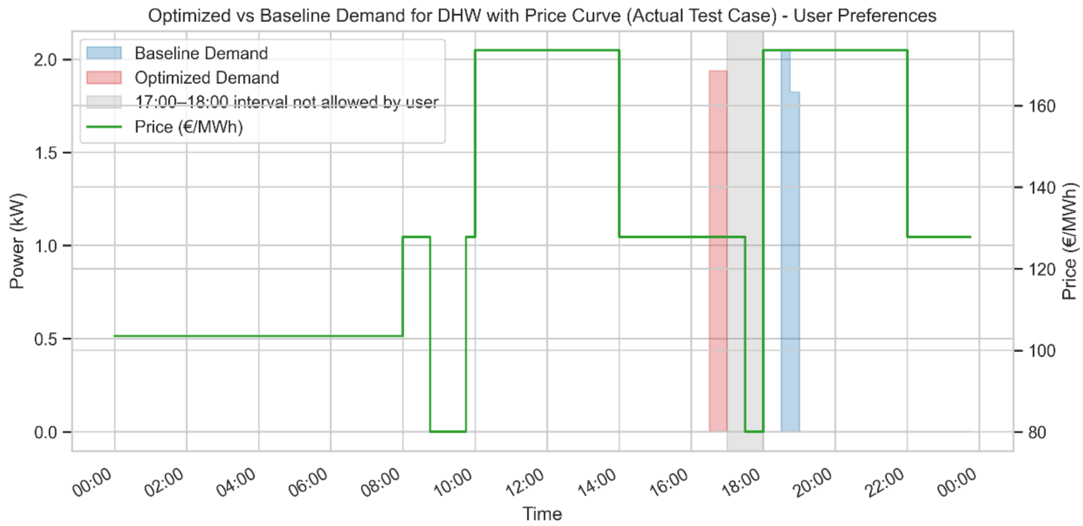


Figure 44: DHW Optimized Profile - Demand shifting allowed by the user and further preferences have been introduced.

The optimized HVAC profile in this scenario is the same as the baseline because this HVAC activation cannot be shifted so much earlier in order to satisfy the constraint that has been set by the user. The user constraint is still satisfied but no demand shifting has taken place.

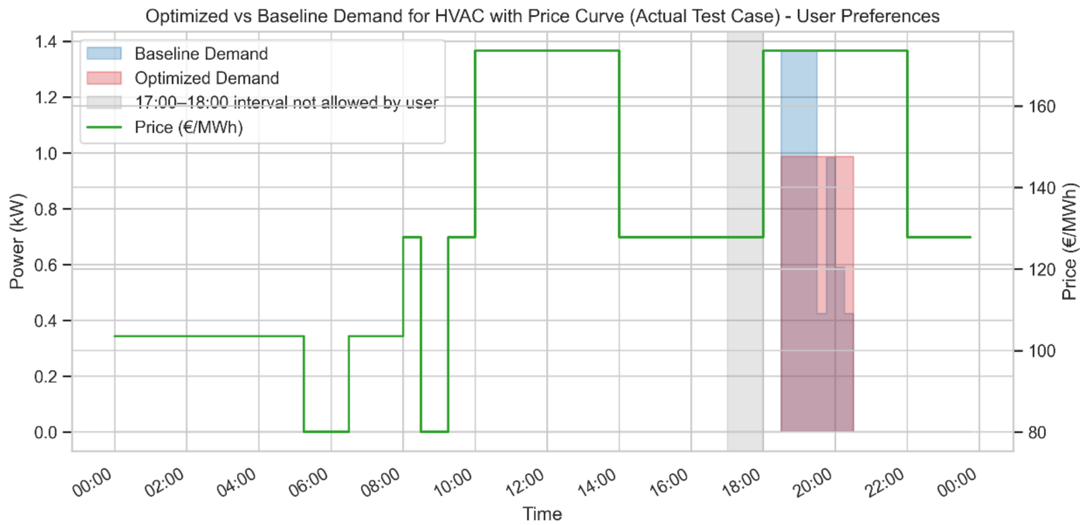


Figure 45: HVAC Optimized Profile - Demand shifting allowed by the user and further preferences have been introduced.

3.7.2.2 Scenarios without Price Profile

3.7.2.2.1. Demand shifting not allowed by the user

The figures in this case show that optimized demand and baseline fully overlap with each other as was the case with prices as well. This is expected since no other option is available except the baseline.

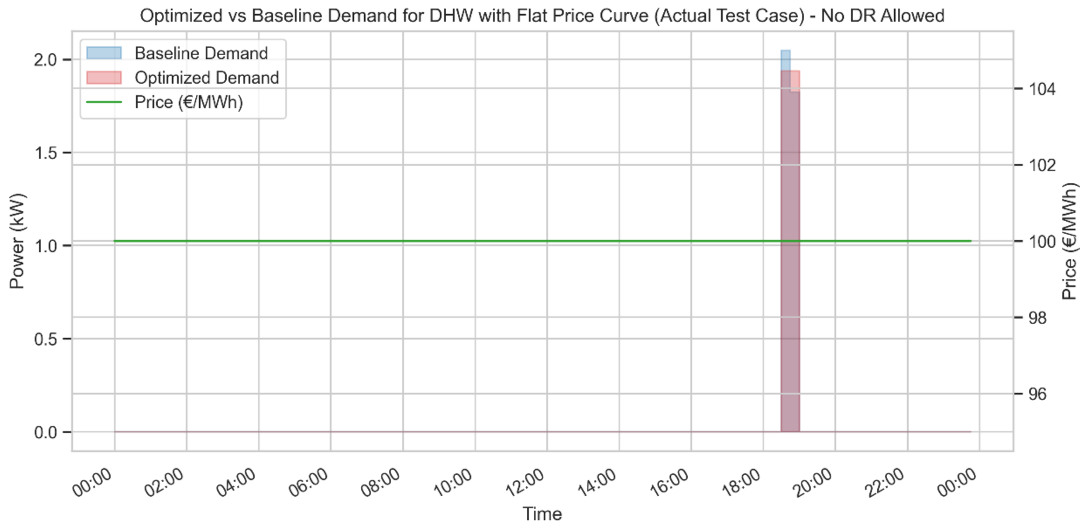


Figure 46: DHW Optimized Profile – Demand shifting not allowed by the user. Price profile not available.

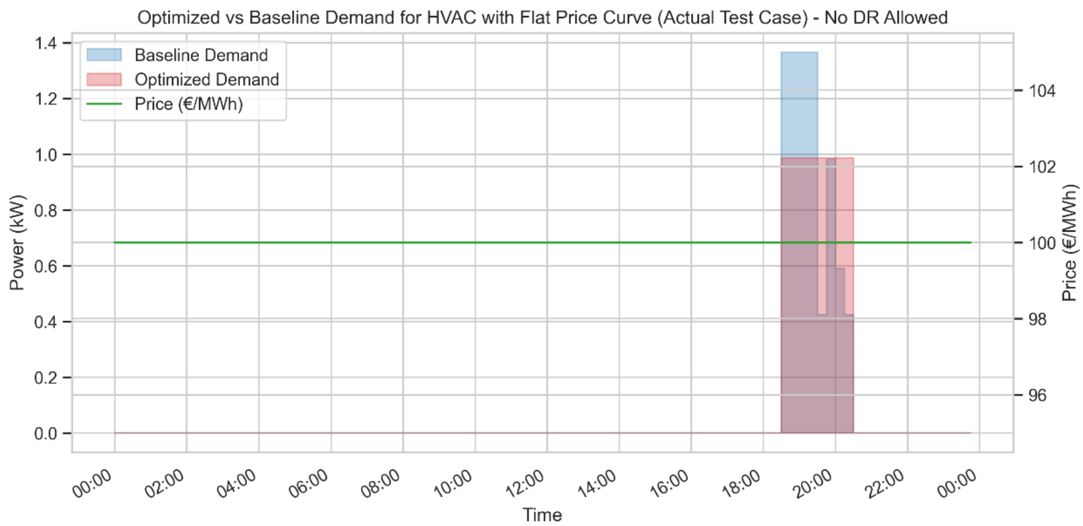


Figure 47: HVAC Optimized Profile – Demand shifting not allowed by the user. Price profile not available.

3.7.2.2.2. Demand shifting allowed by the user and further preferences have not been introduced

Figure 48 shows that DHW demand has been shifted at the earliest possible time interval (16:30 – 17:00), but since price does not influence demand shifting, this activation could have been shifted to any of the following time intervals, 16:45 – 17:15, 17:00 – 17:30, 17:15 – 17:45, 17:30 – 18:00, 17:45 – 18:15, 18:00 – 18:30, which all yield the same objective function value.

Figure 49 shows that HVAC demand has been shifted at the earliest possible time interval. This is because the duration of the activation is equal to two hours which is the same as the flexibility window, therefore, only two options (the one shown in Figure 49 and the baseline) are available no matter whether prices are available or not.

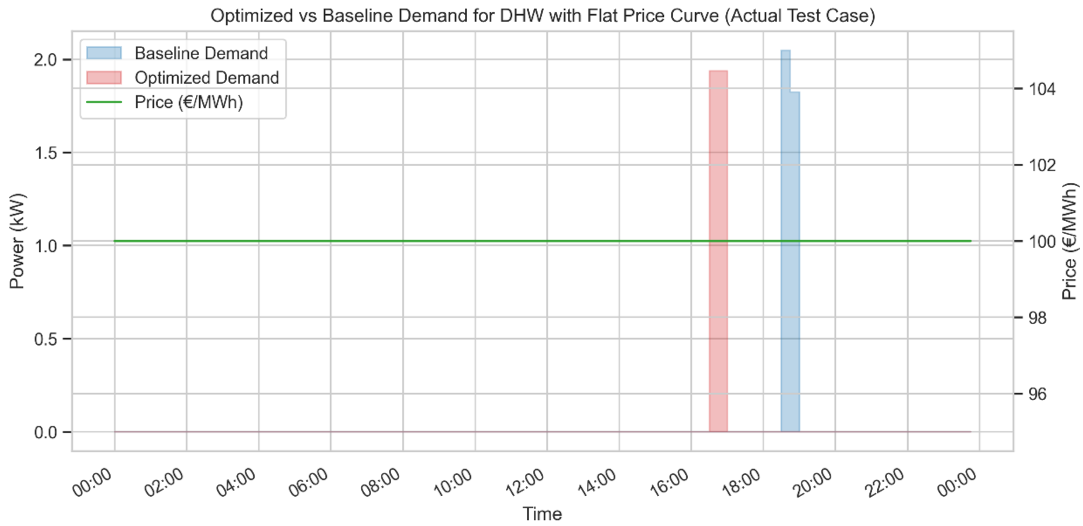


Figure 48: DHW Optimized Profile – Demand shifting allowed by the user and further preferences have not been introduced. Price profile not available.

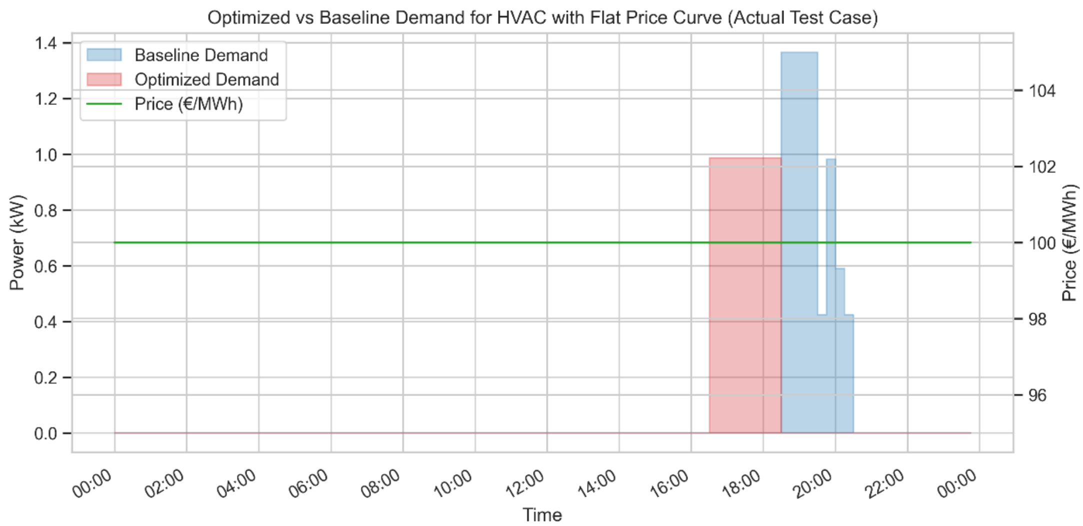


Figure 49: HVAC Optimized Profile – Demand shifting allowed by the user and further preferences have not been introduced. Price profile not available.

3.7.2.2.3. Demand shifting allowed by the user and further preferences have been introduced

Figure 50 shows that DHW demand has been shifted to the earliest possible time interval (16:30 – 17:00). This solution satisfies the constraint set by the user. The activation could also have been shifted to the following time interval: 18:00 – 18:30.

Regarding HVAC demand (Figure 51), the only option allowed, when the user constraint is introduced, is the baseline.

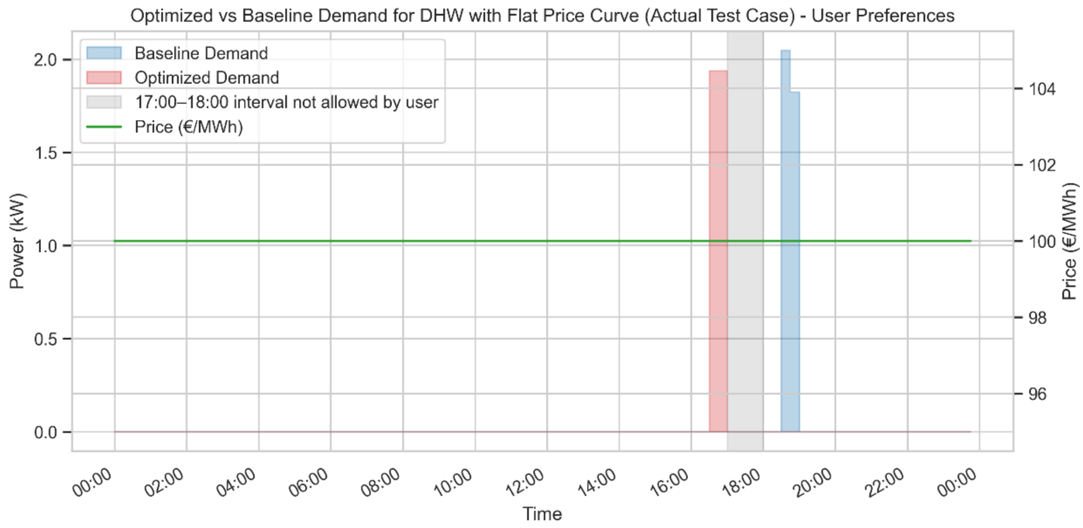


Figure 50: DHW Optimized Profile - Demand shifting allowed by the user and further preferences have been introduced. Price profile not available.

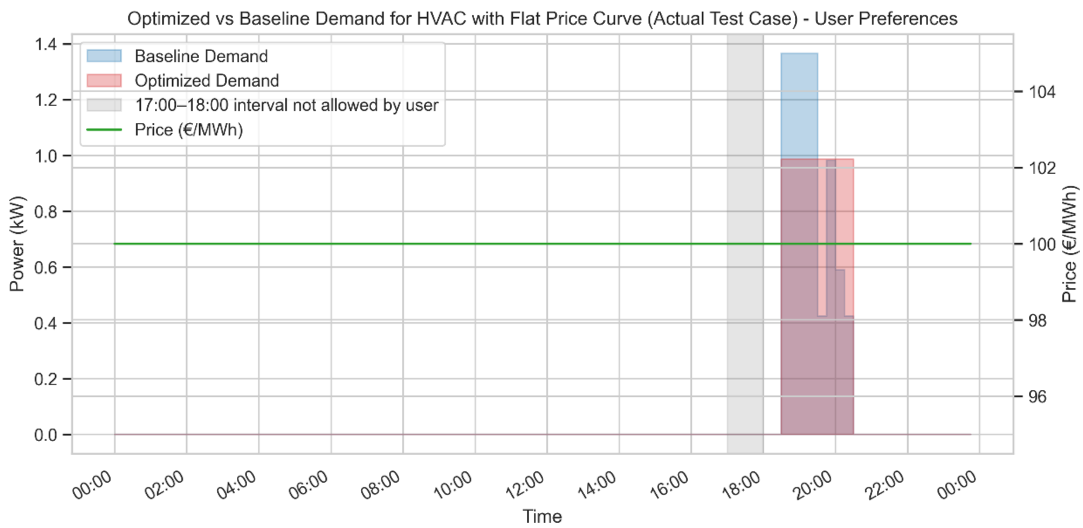


Figure 51: HVAC Optimized Profile - Demand shifting allowed by the user and further preferences have been introduced. Price profile not available.

3.8 Development & integration status

The first version of the deliverable (submitted in M26) provided proof of concept (PoC) of BFMS, using HYPERTECH's data to demonstrate the feasibility and effectiveness of the proposed approach. Between M27 and M34, substantial development efforts were undertaken to mature the system into a fully operational platform, incorporating several improvements and integration activities.

The core BFMS tool underwent extensive optimization and refinement. Algorithmic improvements were implemented to the Flexibility Forecasting Module, along with the finalization and improvement of the DR Initialization Service and the Control Dispatching Service. Issues and bugs identified in the PoC version were also resolved.

In parallel, significant effort was dedicated to integrating the BFMS with external systems. This included comprehensive interfacing with the OPENTUNITY Data Space, as well as the diverse HEMS/BEMS platforms deployed across the pilot sites. Moreover, the integration with the FSP and the NODES market platform was finalized, enabling automated bid generation, submission and retrieval of market results. These integration activities required detailed adaptation of system interfaces, development of middleware components and extensive validation to ensure reliable and secure communication across the aforementioned platforms.

This period also involved iterative testing and verification, using real pilot data to validate the operational performance of the integrated system. Feedback and bugs collected from pilot site leaders lead to further adjustments.

Overall, the work carried out between M27 and M34 reflects a major step forward from the initial PoC, including algorithmic refinements, system updates/enhancements and overall integration, leading to a fully operational, BFMS.

Future identified issues or bugs, potential missing features and feedback received from users during the operation and integration phases will be addressed and resolved within the framework of the demonstration activities.

3.9 Requirements in Equipment & Infrastructure

The successful operation of BFMS relies on the HEMS/BEMS installed at the consumer/prosumer premises and more particularly on the installation of measurement and actuation equipment to retrieve the data necessary for the algorithms training and forecasting as well as for the dispatch of control signals based on the flexibility forecast. This usually involves the installation of the following equipment:

- temperature/humidity sensors,
- occupancy sensors,
- energy meters (total and asset ones, i.e., submetering),
- actuators (control devices)

Ambient and occupant sensors are used for the training of the Comfort Profiling and Building Thermal Modeling Modules, while energy related data are used for the Asset Profiling Module. The actuators are necessary for the successful execution of commands dictated by the incoming DR signals. Each OPENTUNITY pilot site is equipped with a HEMS/BEMs and are capable of satisfying these needs. A detailed description of each HEMS/BEMs is provided in section 2 OPENTUNITY HEMS/BEMS.

3.10 Assumptions and restrictions

The correct BFMS operation highly depends on the reception of all necessary data from the pilot sites and the respective flexibility assets, a necessary step for the training of the flexibility algorithms of the component. Hence, any malfunctions in IoT equipment, such as meters or sensors, can have a significant impact on BFMS functioning, as incomplete metering may lead to inadequately trained models and subsequent inaccurate flexibility forecasts. Similarly, failure to dispatch control commands to the building assets will affect the delivery of estimated flexibility at the pilot sites of the project.

Furthermore, it is important to note that the system's integration and operation have been based on the data streams and conditions encountered up to this point of the project. While the core development and integrations are complete, a full-scale deployment across all pilots may reveal unforeseen technical issues, or testing scenarios not previously observed. Continuous monitoring and potential adjustments might be necessary to ensure a reliable performance in a real-world operational environment.

4. NILM and Behaviour Analytics

4.1 Component overview

As electricity prices continue to rise, optimizing electricity consumption at economic and energy level throughout the value chain is essential. Energy tariffs are constrained by the energy market, offering limited profit margins. On the other hand, while people are interested in appliance energy consumption, they lack tools to access energy-related information. This is where Non-Intrusive Load Monitoring (NILM) becomes relevant.

NILM is a methodology that detects appliance usage based on total power or energy consumption curves. It identifies patterns using exogenous variables such as active power, reactive power, current, voltage, and energy consumption, disaggregating the power usage of each appliance from the total consumption curves.

NILM has been extensively researched, with numerous algorithms proposed [1]. Initial approaches included linear optimizations based on maximum appliance power consumption and event detection [2] [3], Hidden Markov Models for pattern recognition approaches [4] and Python toolkits for developing and training various mathematical algorithms were also developed [5].

Recently, the AI boom has introduced machine learning and deep learning methodologies, yielding promising results. Supervised learning techniques, which use real data for model evaluation, have shown the best performance. Classification techniques predict appliance usage in binary terms (on/off) using models like Random Forest, Support Vector Machines, and Neural Networks [6]. Regression techniques predict numerical power consumption using models like XGBoostRegressor and Long Short-Term Memory neural networks [7]. However, supervised methods require labeled data, necessitating intrusive load monitoring or user data tagging. To prevent this, unsupervised and semi-supervised methods, such as k-means clustering [8] [9], have been developed to detect patterns and determine appliance usage with the cost of reduced model performance.

This project employs regression techniques to predict real-time individual appliance power consumption, creating generalized models applicable to any household with metering data. This approach clarifies electrical measurement information for end-users, indicating appliance performance more accurately. Additionally, these generalized models support demand response campaigns by providing consumption curves for flexible assets.

4.2 Component Prototype

The architecture defined in D2.3 with the SGAM framework can be seen in Figure 52. Component layer is shown, with the different components involved.

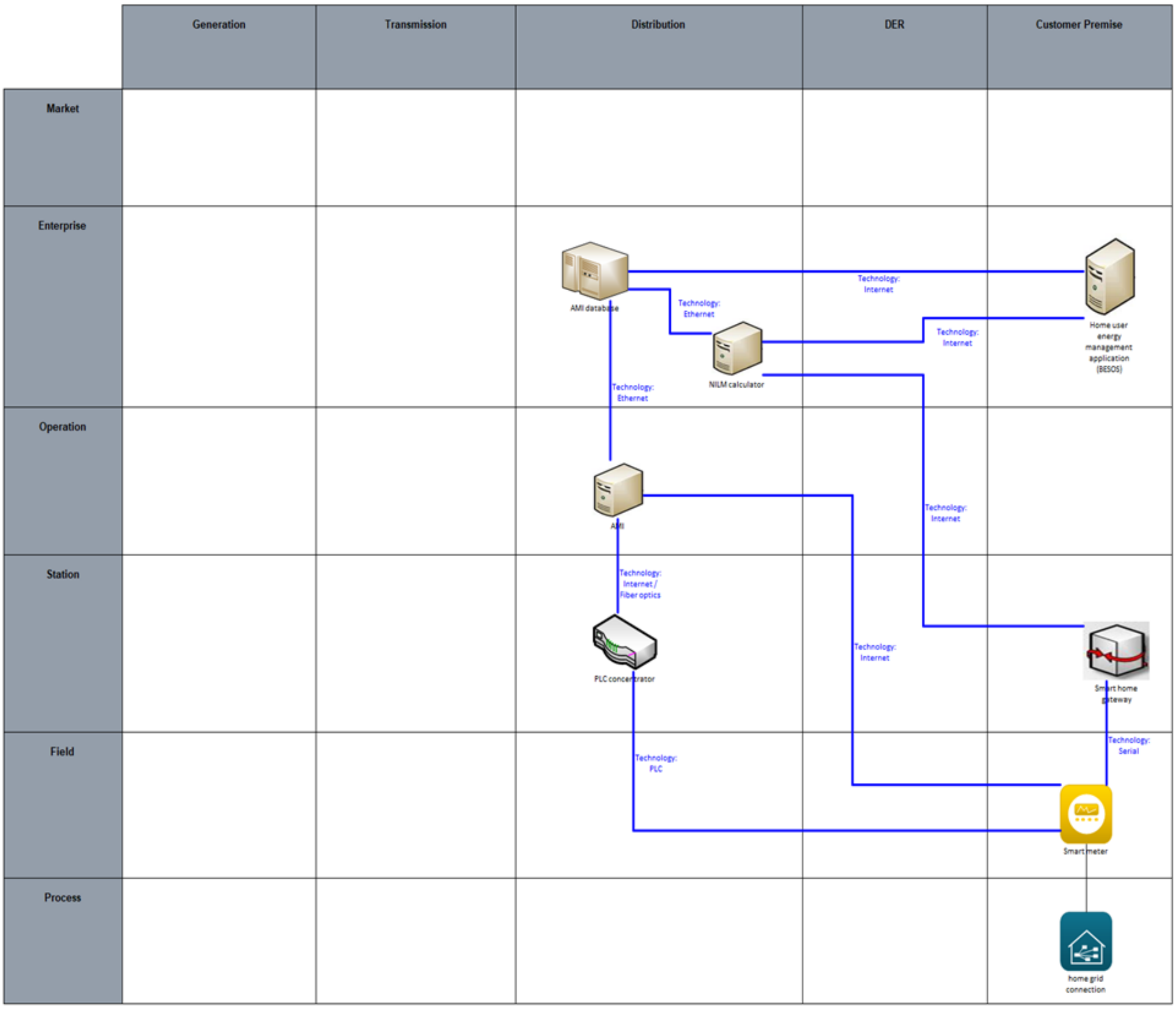


Figure 52: NILM and behavior analytics SGAM component layer

The central component of the NILM Use Case. is the NILM calculator. A subcomponent diagram can be seen in Figure 53 and is defined as follows:

- **Metering data:** This includes all information from smart meters and submetering in households selected from the pilots. Data is retrieved from their respective databases and Dataspace connectors defined in the project. It includes electrical measurements such as active and reactive power, current, voltage, etc.
- **API / Dataspace connector:** Gateways to retrieve information, including real-time measurements and historical data necessary to train the AI models. The primary aim is to use Dataspace connectors, but a database API can also be used to retrieve information directly.
- **NILM calculator:** This module retrieves real-time electrical measurement data to predict the individual power consumption of each appliance and sends this information back to the Dataspace connector or the corresponding database. The prediction models are stored and downloaded from the model storage software.
- **Model Storage:** Software designed to store AI models along with their evaluation metrics, datasets, and graphical content, including versioning and deployment information. Models

are trained offline using historical and publicly available data. Uploading and downloading of models are managed via the corresponding API.

- **NILM GUI:** This graphical user interface displays all electrical measurement information, including real-time, historical, and prediction data, using a dashboard. It is organized in different pages:
    - **Dashboard:** shows KPIs and plots regarding consumption data in real time and historical aggregated into different periods.
    - **Certificates:** users can input specific appliance information such as brand and model, and through an API, both theoretical and current energy certificates will be displayed.
    - **Events:** shows events regarding predictions performed over historical data. Then, the user can provide feedback for whether the predictions were correct or not.
    - **Feedback**: provides a form to gather feedback regarding the application and its usage.
- **NILM certificates:** This module calculates energy certificates based on the selected appliance and various input parameters. It computes a theoretical certificate and, using historical data, generates a certificate based on actual energy usage, allowing for comparison.

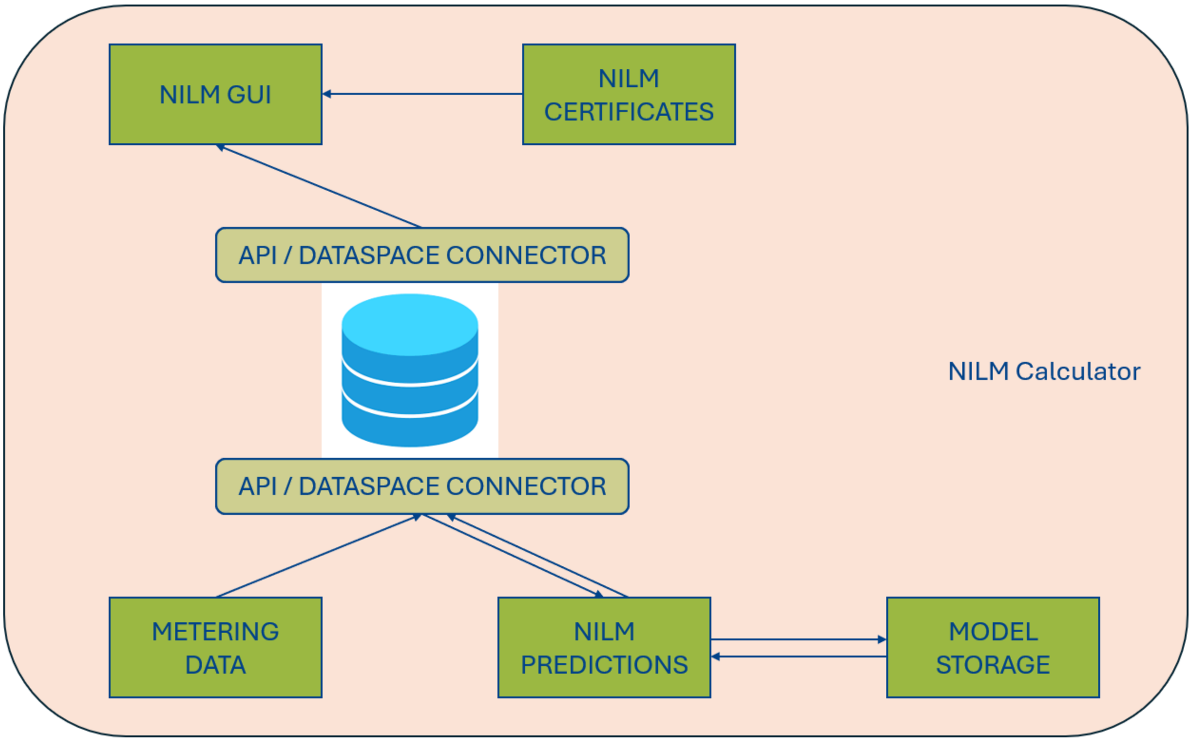


Figure 53: NILM calculator detailed subcomponent architecture.

The GUI is componentized, meaning it can be integrated into other software systems. Specifically, it has been added to the HEMS software of ETRA called ELIOT.

Components and subcomponents are designed for cloud deployment, allowing resource scaling as needed. The architecture utilizes microservices and Docker, enabling independent refinement of each subcomponent and simplifying deployments. Communication between subcomponents is managed via API, following the microservice architecture, while communication with pilot data is handled using Dataspace connectors.

4.3 Proposed Methodology

The methodology selected for NILM calculations involves supervised machine learning methods. The goal is to train generalized models for each appliance using historical data from publicly available datasets and the project's pilot data. This methodology requires Intrusive Load Monitoring for training, validation, and evaluation, but once trained, the models can be applied to any household.

The datasets available online consist of measurements from various households worldwide. Research papers have attempted to consolidate these datasets into a common repository [10]. These datasets vary in granularity, period, number of households, and collected electrical measurements, allowing for the creation of diverse machine learning models based on data needs. The primary benefit of this method is that generalized models can be applied to any household providing total energy or power consumption. Additional electrical measurements can enhance model performance, and submetering data can re-train models for increased accuracy.

Machine learning models are trained using collected information. The target variable is the individual power consumption of each appliance. Models have been trained using only active power consumption and both active and reactive power consumption as exogenous variables due to the lack of datasets with more than active and reactive power included and other electrical-related information such as voltage or current. Cross-validators and hyperparameter optimizers, such as BayesianSearchCV and RandomSearchCV, are used to determine optimal hyperparameters, minimizing prediction errors.

The optimizers need the numeric or class intervals to search for each hyperparameter, using a randomized search and trying to reduce the evaluation metric desired, the optimizator starts trying out some models with random hyperparameters in the defined grid for a defined number of iterations. Once the iterations are completed, the best performing model is served. The cross validators used are BayesianSearchCV and RandomSearchCV.

The main packages used for modeling are XGBoost and TensorFlow. XGBoost, short for Extreme Gradient Boosting, is a powerful and efficient implementation of gradient-boosted decision trees. It is popular for its fast learning and inference times, making it one of the best-performing models for structured/tabular data. The model works by sequentially adding decision trees, where each new tree corrects the errors of the previous ones. This boosting process helps to improve accuracy and reduce overfitting. XGBoost also includes advanced features like tree pruning, regularization, and parallel processing to enhance performance.

TensorFlow, on the other hand, is an open-source deep learning framework developed by Google. It is widely used for building and training neural networks. TensorFlow supports various types of neural network architectures, including Convolutional Neural Networks (CNNs) for image processing, Recurrent Neural Networks (RNNs) for sequential data, and fully connected networks for general-purpose tasks. The training process in TensorFlow involves optimizing the weights of the neural network using algorithms like stochastic gradient descent (SGD) and its variants (e.g., Adam, RMSprop). TensorFlow's flexibility and scalability make it suitable for both research and production environments.

In addition to the supervised prediction stage, a post-prediction analytics module is incorporated to further improve the accuracy and interpretability of the appliance-level disaggregation. Once the machine learning models have predicted the estimated power consumption per appliance, these predictions are analyzed collectively against the observed aggregate power signal of the household to refine and validate the disaggregation results.

The first step in this process involves detecting significant consumption spikes in the household's total active power curve. These spikes are defined based on a rolling quantile-based smoothing function and dynamic threshold conditions. For each detected spike, the algorithm analyzes which appliance predictions best explain the energy observed during the event. If the combined predictions of candidate appliances account for a sufficient proportion of the energy within the spike, and one appliance stands out distinctly (according to confidence thresholds), the event is attributed to that appliance. This attribution is governed by relative energy ratios and a tolerance margin to ensure robustness. Subsequently, predictions of unrelated appliances during the same time window are suppressed, resulting in a corrected power series.

For spikes not confidently explained in the first pass, a second step is applied. A similarity-based classification mechanism compares these unclassified events with previously labelled and explained events. Features such as spike duration, total energy, and power statistics (mean, max, min) are used to compute cosine similarity over a normalized feature space. If a sufficiently similar event is found, the unmatched spike is reclassified accordingly, enhancing disaggregation coverage.

Unresolved spikes after both passes are defined as unidentified events. To further improve the system, a graphical user interface (GUI) functionality has been developed. This interface allows users to inspect each detected spike, view the predicted appliance attribution, and manually correct or label those cases that were incorrectly classified or left unassigned. This user feedback mechanism not only improves the transparency of the system's outputs but also enables potential re-training of models in future iterations.

MLflow open-source software is deployed to track the progress of each trained model. Evaluation metrics, datasets, trained models, and performance plots are available for each training run. MLflow also supports versioning and model downloading via API.

Regarding energy certificates, the European Commission defines a common procedure to calculate the label for each appliance [11]. These labels often involve annual or per-cycle energy consumption values, calculated during testing periods by manufacturers under specific conditions. These theoretical values can be compared with real values from submetering or NILM calculations, providing insights into appliance performance using energy labeling.

4.4 Technology stack and Implementation tools

The technologies selected can be divided between the different components:

Table 6: Libraries and Technologies used in NILM predictions

Library/Technology Name	Version	License
Python	3.10.12	PSF License Agreement and the Zero-Clause BSD license

Pandas	2.0.3	BSD 3-Clause "New" or "Revised" License
Numpy	1.24.4	Modified BSD license
Matplotlib	3.7.5	PSF license
xgboost	2.0.3	Apache License 2.0
tensorflow	2.10	Apache License 2.0
Scikit-learn	1.3.2	BSD 3-Clause "New" or "Revised" License
influxdb	5.3.2	MIT license
mlflow	2.11.1	Apache License 2.0
schedule	1.2.1	MIT license

Table 7: Libraries and Technologies used in Model Storage

Library/Technology Name	Version	License
MLflow	2.13.0	Apache License 2.0

Table 8: Libraries and Technologies used in NILM GUI

Library/Technology Name	Version	License
Zustand	4.1.1	MIT License
React-router-dom	6.22.2	MIT License
React-i18next	12.2.0	MIT License
React-favicon	2.0.3	MIT License
React-dom	18.2.0	MIT License
React	18.2.0	MIT License
Notistack	3.0.1	MIT License
Natsex	2.0.16	MIT License
Moment	2.29.4	MIT License
Meteor-node-stubs	1.2.10	MIT License
Lodash	4.17.21	MIT License

I18next-browser-languagedetector	6.1.5	MIT License
I18next	21.9.2	MIT License
Etra-table-material	0.0.21	MIT License
Etra-mui-theme	2.0.4	MIT License
Etra-mui-components	2.0.1	MIT License
Etra-metrics-components	4.0.147	MIT License
Citric-helper	1.5.17	MIT License
@mui/styles	6.1.2	MIT License
@mui/material	5.15.19	MIT License
@mui/icons-material	5.15.19	MIT License
@fontsource/roboto	5.0.8	MIT License
@emotion/styled	11.13.0	MIT License
@emotion/react	11.11.4	MIT License
@babel/runtime	7.24.0	MIT License

Table 9: Libraries and Technologies used in NILM certificates

Library/Technology Name	Version	License
fastapi	0.115.5	MIT license
pandas	2.2.3	BSD 3-Clause "New" or "Revised" License
numpy	2.2.1	Modified BSD license
requests	2.32.3	Apache License 2.0
influxdb	5.3.2	MIT license
Uvicorn	0.32.1	BSD 3-Clause "New" or "Revised" License
Python	3.10.12	PSF License Agreement and the Zero-Clause BSD license

4.5 Input/Output parameters

The inputs and outputs necessary to make use of the components are:

Table 10: Inputs required, and outputs delivered by NILM predictions

Inputs (* optional)	Units
Active power (Total from the household)	Watts
Reactive power (Total from the household) (*)	Volts-Ampere reactive
Current (*)	Ampere
Voltage (*)	Volts
List of appliances inside the household	Washing machine, dryer, air conditioner, water heater, fridge, oven, etc.
Outputs	Units
Active power (Individual consumption of each appliance monitored)	Watts

Table 11: Inputs required, and outputs delivered by Model Storage

Inputs	Units
Model name	-
Model version	-
Appliance	-
Outputs	Units
Machine-learning model class	-

Table 12: Inputs required, and outputs delivered by NILM GUI

Inputs (* optional)	Units
Active power (Total from the household)	Watts
Reactive power (Total from the household) (*)	Volts-Ampere reactive
Current (*)	Ampere
Voltage (*)	Volts
List of appliances inside the household	Washing machine, dryer, air conditioner, water heater, fridge, oven, etc.
Active power (Individual consumption of each appliance monitored)	Watts
Outputs	Units
-	-

Table 13: Inputs required, and outputs delivered by NILM Certificates disaggregated into the different appliances

Fridge and Freezer inputs	Units
Annual energy consumption	kWh/annum
Number of doors	-
Design type	Free-standing or built-in
Compartment type	Fresh food, wine storage, etc.
Compartment volume	dm^3 or litres
Compartment defrosting type	Auto-defrost or manual defrost
User	-
Fridge and Freezer outputs	Units
Theoretical energy efficiency index	-
Current energy efficiency index	-
Theoretical energy label	A to G
Current energy label	A to G
Washing machine inputs	Units
Rated capacity	Kg
Energy consumption per cycle, eco 40-60 programme	kWh
User	-
Theoretical energy efficiency index	-
Current energy efficiency index	-
Theoretical energy label	A to G
Current energy label	A to G
Washer-dryer inputs	Units
Rated capacity (complete cycle)	Kg
Energy consumption per cycle (complete cycle)	kWh
Rated capacity (Washing cycle)	Kg

Energy consumption per cycle (washing cycle)	kWh
User	-
Washer-dryer outputs	Units
Theoretical energy efficiency index (complete cycle)	-
Current energy efficiency index (complete cycle)	-
Theoretical energy label (complete cycle)	A to G
Current energy label (complete cycle)	A to G
Theoretical energy efficiency index (washing cycle)	-
Current energy efficiency index (washing cycle)	-
Theoretical energy label (washing cycle)	A to G
Current energy label (washing cycle)	A to G
Dishwasher inputs	Units
Assigned capacity	Liters
Width	Centimetres
Energy consumption (per cycle, based on the eco programme)	kWh
User	-
Theoretical energy efficiency index	-
Current energy efficiency index	-
Theoretical energy label	A to G
Current energy label	A to G
Dryer inputs	Units
Rated capacity	Kg
Weighted Annual Energy Consumption	kWh/annum
User	-
Dryer outputs	Units
Theoretical energy efficiency index	-

Current energy efficiency index	-
Theoretical energy label	A^{+++} to D
Current energy label	A^{+++} to D
Air conditioner inputs	Units
Seasonal Energy Efficiency Ratio (SEER)	-
Annual Electricity Consumption (Cooling mode)	kWh/annum
Seasonal Coefficient of Performance (SCOP)	-
Annual Electricity Consumption (Heating mode)	kWh/annum
User	-
Air conditioner outputs	Units
Theoretical energy efficiency index (SEER)	-
Current energy efficiency index (SEER)	-
Theoretical energy label (SEER)	A^{+++} to G
Current energy label (SEER)	A^{+++} to G
Theoretical energy efficiency index (SCOP)	-
Current energy efficiency index (SCOP)	-
Theoretical energy label (SCOP)	A^{+++} to G
Current energy label (SCOP)	A^{+++} to G
Water Heater inputs	Units
Load profile	3XS to XXL
Water heating energy efficiency	percentage
Annual electricity consumption	kWh
User	-
Water Heater outputs	Units
Theoretical energy efficiency index	-
Current energy efficiency index	-
Theoretical energy label	A^{+++} to G
Current energy label	A^{+++} to G

Television inputs	Units
On mode power demand in Standard Dynamic Range (SDR)	Watts
Visible screen area	dm^2
User	-
Television outputs	**Units**
Theoretical energy efficiency index	-
Current energy efficiency index	-
Theoretical energy label	A to G
Current energy label	A to G
Oven inputs	**Units**
Oven volume	Liters
Energy consumption per cycle (fan-forced convection mode)	kWh
User	-
Oven outputs	**Units**
Theoretical energy efficiency index	-
Current energy efficiency index	-
Theoretical energy label	A^{+++} to D
Current energy label	A^{+++} to D

4.6 Interfaces & API documentation

To clearly understand the data flows, refer to Figure 54. The goal is to have all information available in the OPENTUNITY Data Space using the Data Space Connector.

First, raw electrical measurements from households should be available to request total values and the list of connected appliances from the NILM predictions tool. AI models are then downloaded from the Model Storage using its Python API client, see [12]. This model storage is created using MLflow, which provides a GUI for performing actions that can also be done through the API, including run modifications, model registrations, results visualizations, etc.

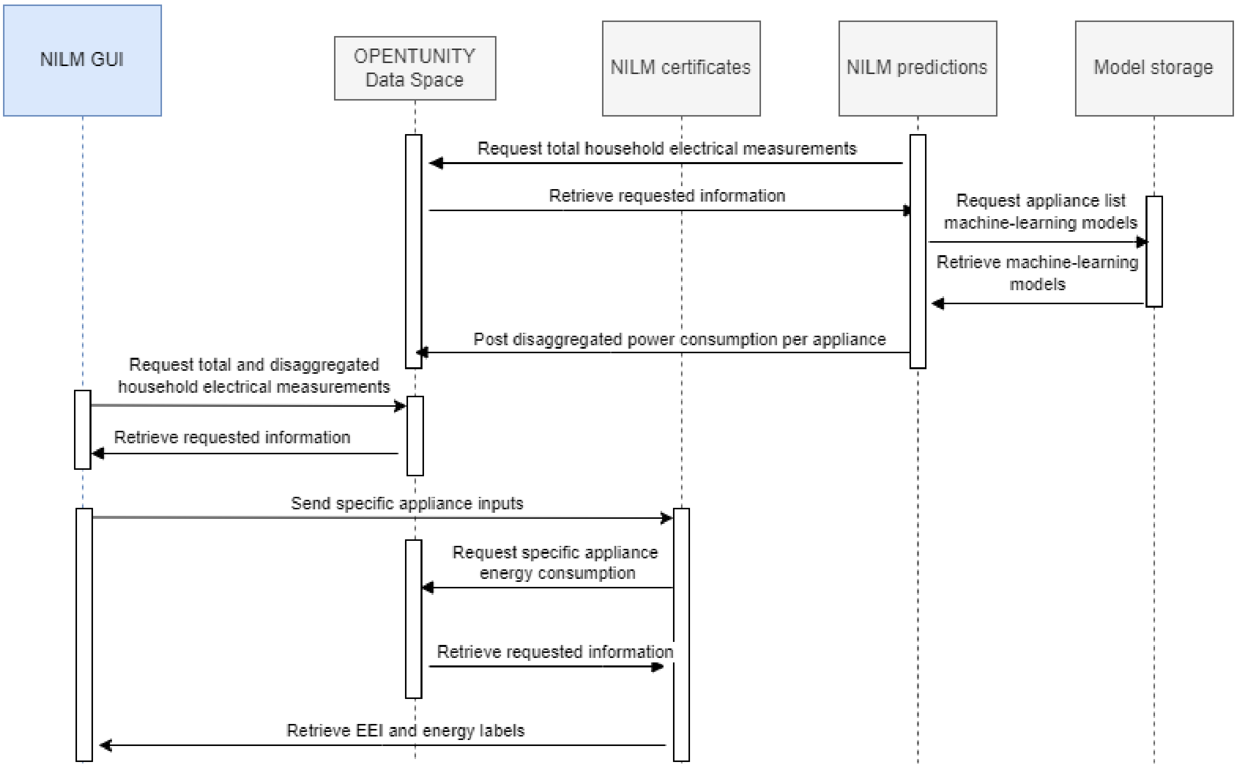


Figure 54: NILM calculator dataflows diagram.

All machine-learning models are stored in "runs", see Figure 55. These runs save evaluation metrics, models, datasets used, and tags to track each model's performance. Once runs are completed and compared, the best-performing models can be registered and versioned using the UI or API for future download if necessary.

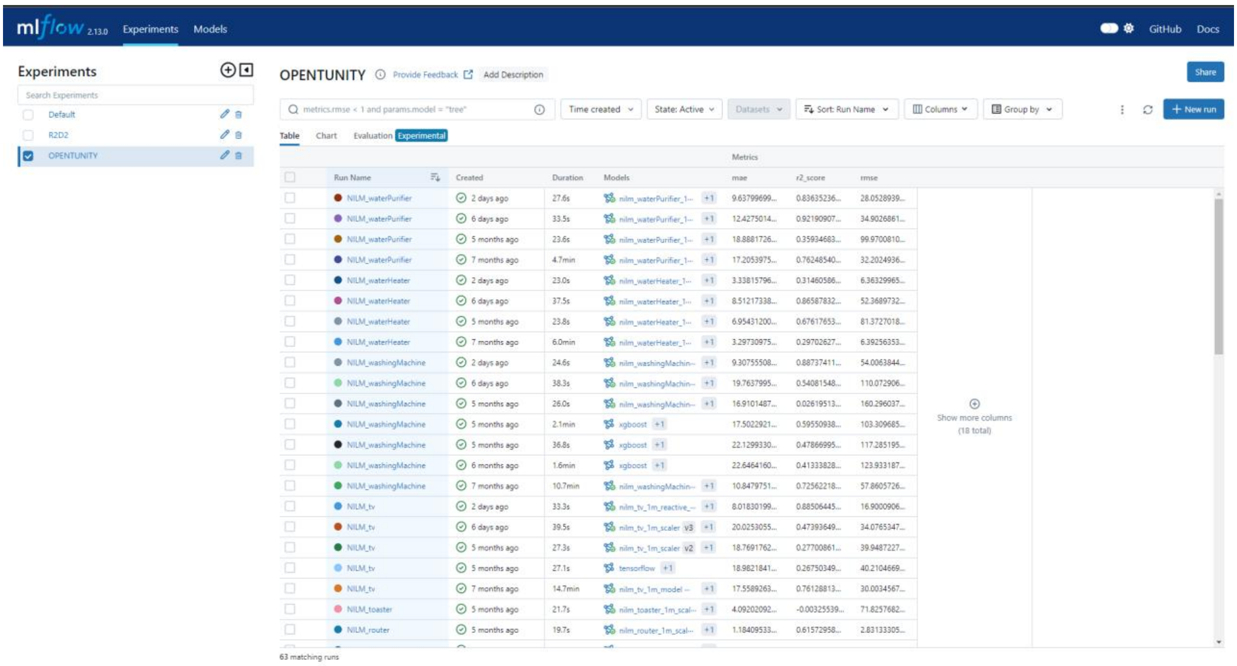


Figure 55: MLflow graphical user interface, runs page.

The models are downloaded into the NILM calculator module, which acts as an API for appliance disaggregation. This API has a Swagger/OpenAPI specifications, see Figure 56. It exposes a single POST endpoint, /predict, which receives a request body containing the input parameters described in Table 14. The format and structure of a typical response are shown in Figure 57. The Dataspace acts as a proxy to this API, meaning that the URL providing the endpoints are held by the Dataspace connector.

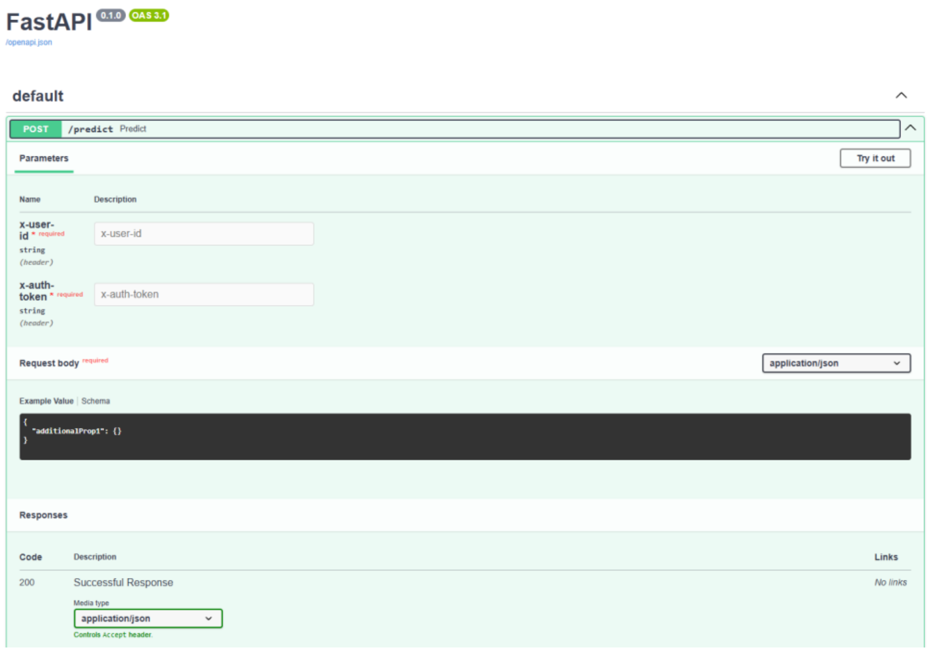


Figure 56. NILM calculator API, swagger documentation.

Parameter	Type	Description
data	Array of objects	Each object should include timestamp, activePower, and optionally reactivePower or apparentPower. If data_type is "active", only activePower is required. If "reactive", reactivePower or apparentPower must also be provided.
appliances	Array of strings	Appliances available: "dishwasher", "fridge", "microwave", "washingMachine", "dryer", "kettle", "waterHeater", "oven", "tv"
granularity	String	Accepts "1m" or "15m"
data_type	String	"active" or "reactive" – defines which power data will be used for disaggregation.

Table 14. NILM calculator API parameter description.

```json
{
  "data": [
    {
      "timestamp": "2023-10-01T00:00:00Z",
      "fridgeActivePower": 100,
      "washingMachineActivePower": 200,
      "dishwasherActivePower": 150,
      "dryerActivePower": 50
    },
    {
      "timestamp": "2023-10-01T00:01:00Z",
      "fridgeActivePower": 110,
      "washingMachineActivePower": 210,
      "dishwasherActivePower": 160,
      "dryerActivePower": 60
    }
  ],
  "n_lags": 180,
  "details": "Prediction completed successfully"
}
```


Figure 57. NILM calculator API response example.

At this point, the GUI is responsible for gathering the total consumption measurements and interact with the API to obtain the disaggregated results., and a series of KPIs and metrics integrated into the GUI can be displayed, as shown in Figure 59 and Figure 60.

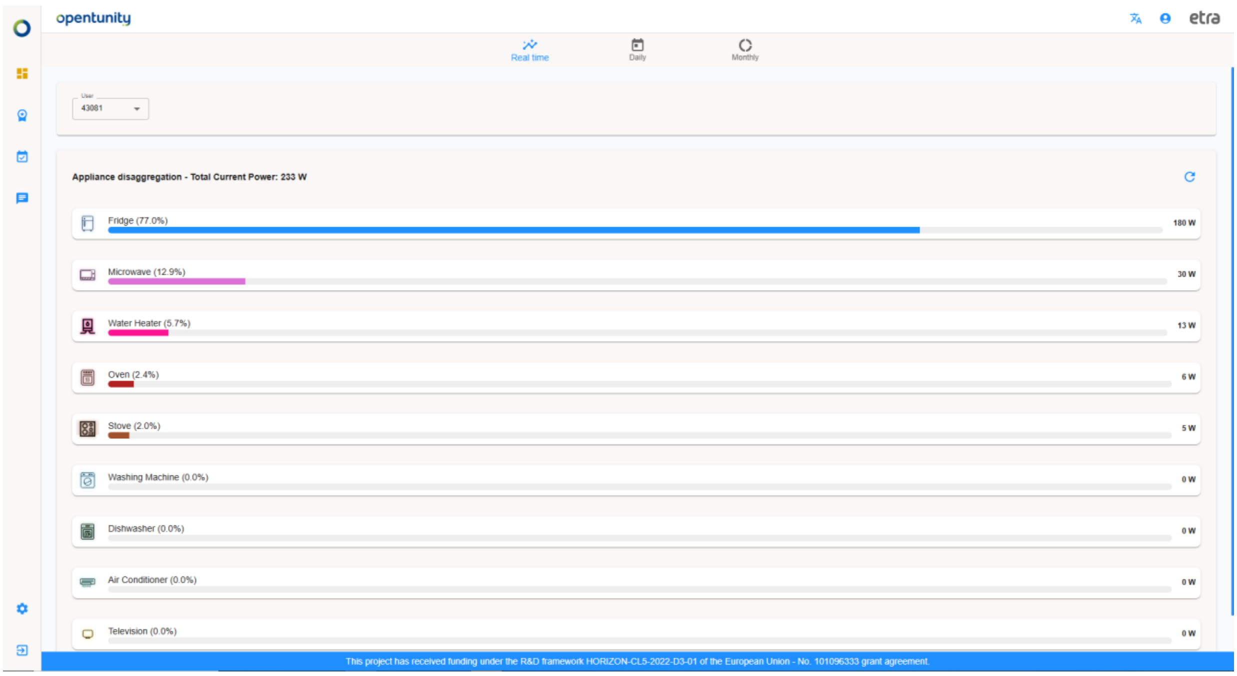


Figure 58. NILM graphical user interface, dashboard page. Real-time metrics.

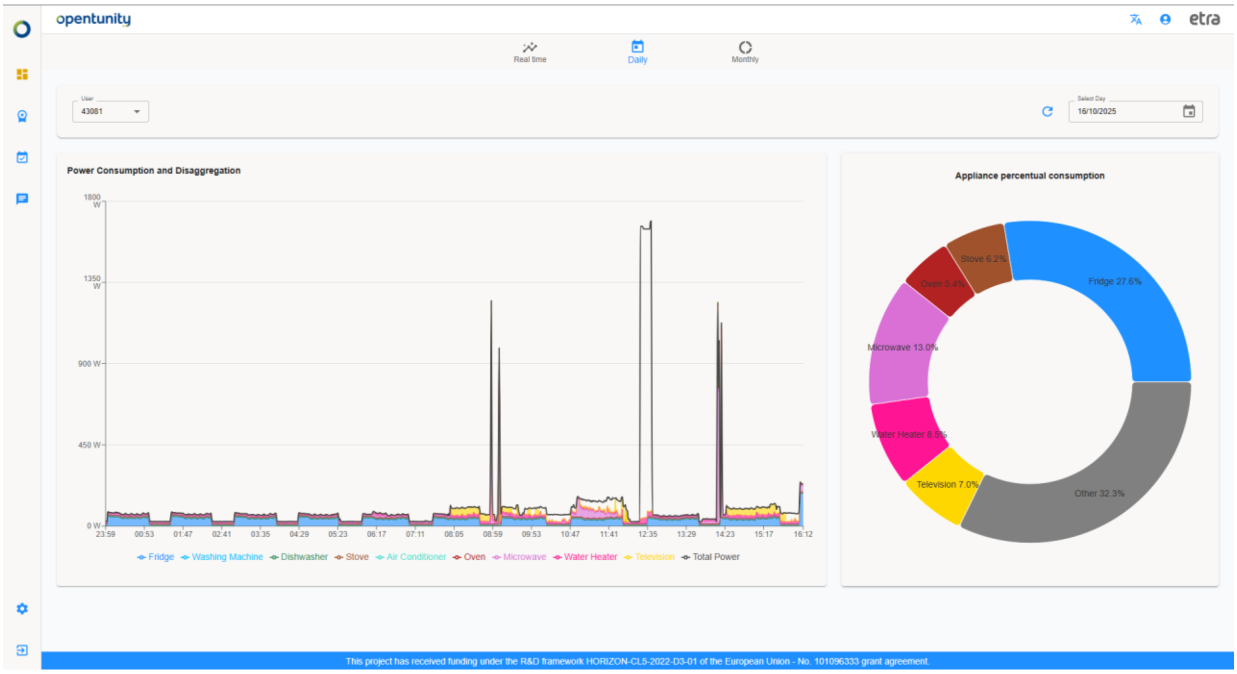


Figure 59: NILM graphical user interface, dashboard page. Daily metrics.

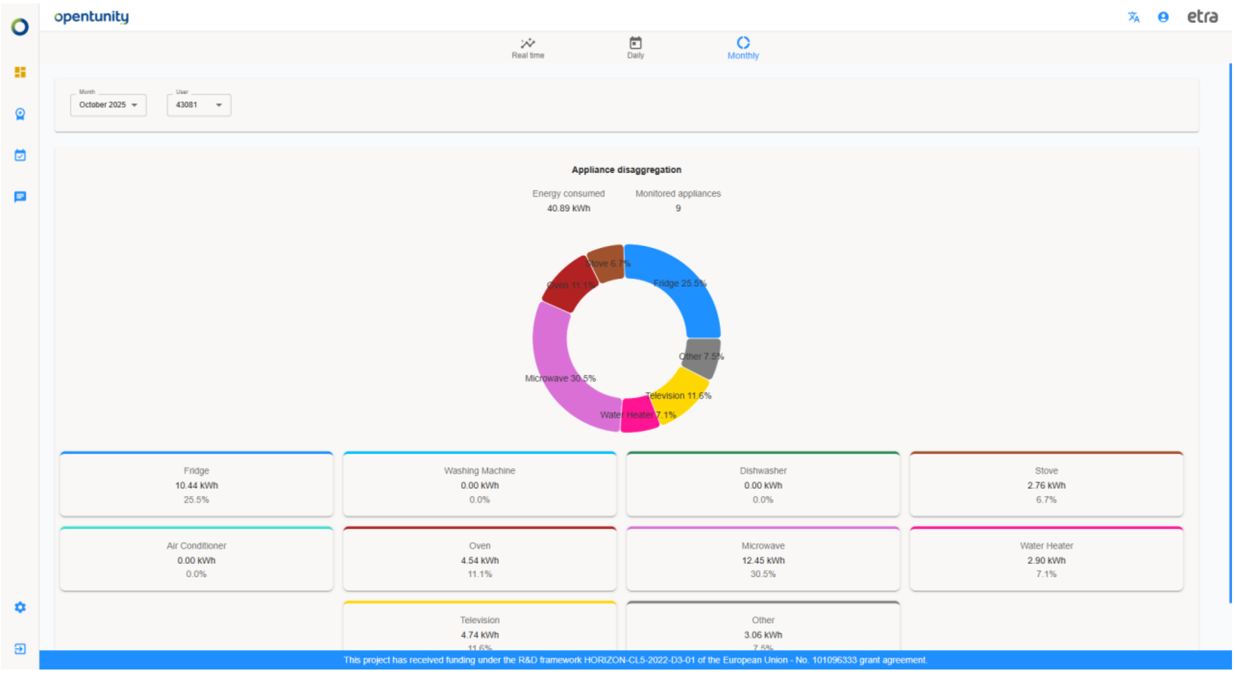


Figure 60. NILM graphical user interface, dashboard page. Monthly metrics.

The energy certificates page is shown in Figure 61. One energy certificate is already calculated, but to generate one, a series of inputs are needed from the end-user through the configuration tab to find the corresponding appliance brand and model. These inputs are then used to find the appliance specifications in the European Product Registry for Energy Labelling (EPREL) database. Once completed, the information is sent to the NILM Certificates API, which performs calculations using both theoretical data and real data from the household. Both theoretical and current energy labels, along with their corresponding Energy Efficiency Index (EEI), are displayed. The API uses the POST method with the appliance name, user ID, and corresponding characteristics in JSON format to create the certificate. The results are also sent in JSON format with the EEI and energy label, as seen in Figure 61.

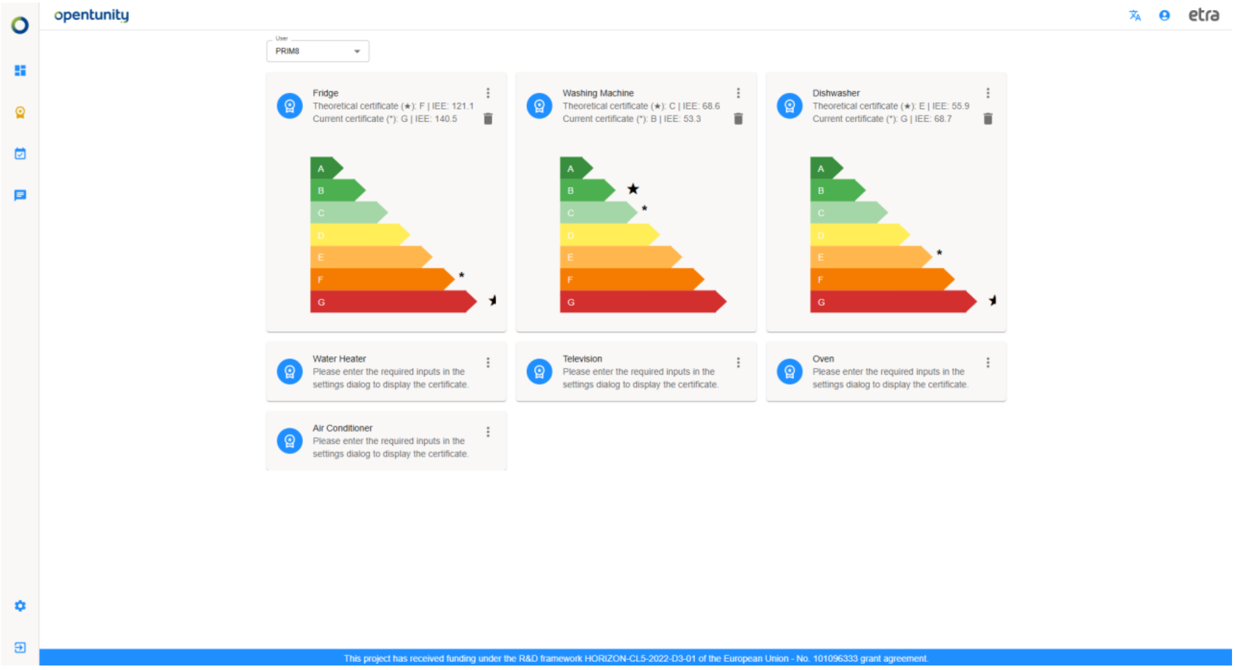


Figure 61: NILM graphical user interface, certificates page.

The Events page provides a detailed interface for reviewing predicted appliance activations throughout the latest day or a selected custom date. As shown in Figure 62, this section lists all detection events generated during the prediction and post-processing stages described earlier. Each listed event displays the estimated appliance responsible for the detected peak and the time interval when the event occurred. To ensure transparency and allow manual validation of the system outputs, each event entry includes a user feedback form.

This built-in form allows the user to either confirm the appliance identified by the NILM system or manually select the correct appliance if the prediction is inaccurate or if no detection was made. An optional text input field is also provided to include any additional user comments or contextual information, allowing for more detailed feedback. All user feedback from this form is stored in the database for further analysis and potential retraining use.

Lastly, to complement this functionality, a dedicated Feedback page is also integrated into the graphical interface, see Figure 63 and Figure 64. This page is designed to gather user impressions related to the overall application, such as usability, performance, clarity of results, and satisfaction with recommendations. It presents a direct link to a feedback form deployed by Joanneum Research via a specialized platform called Green Experience Lab for user surveys and analytics. This form enables user engagement monitoring, collection of structured opinions, and identifying recurring issues or feature requests.

The NILM application is optimized for mobile devices. All visualizations and forms, including the Events and Feedback pages, can be accessed via smartphones or tablets with consistent functionalities. This ensures that users can access from any location, promoting user engagement and a more flexible interaction with the interface.

Figure 62: NILM graphical user interface, events page.

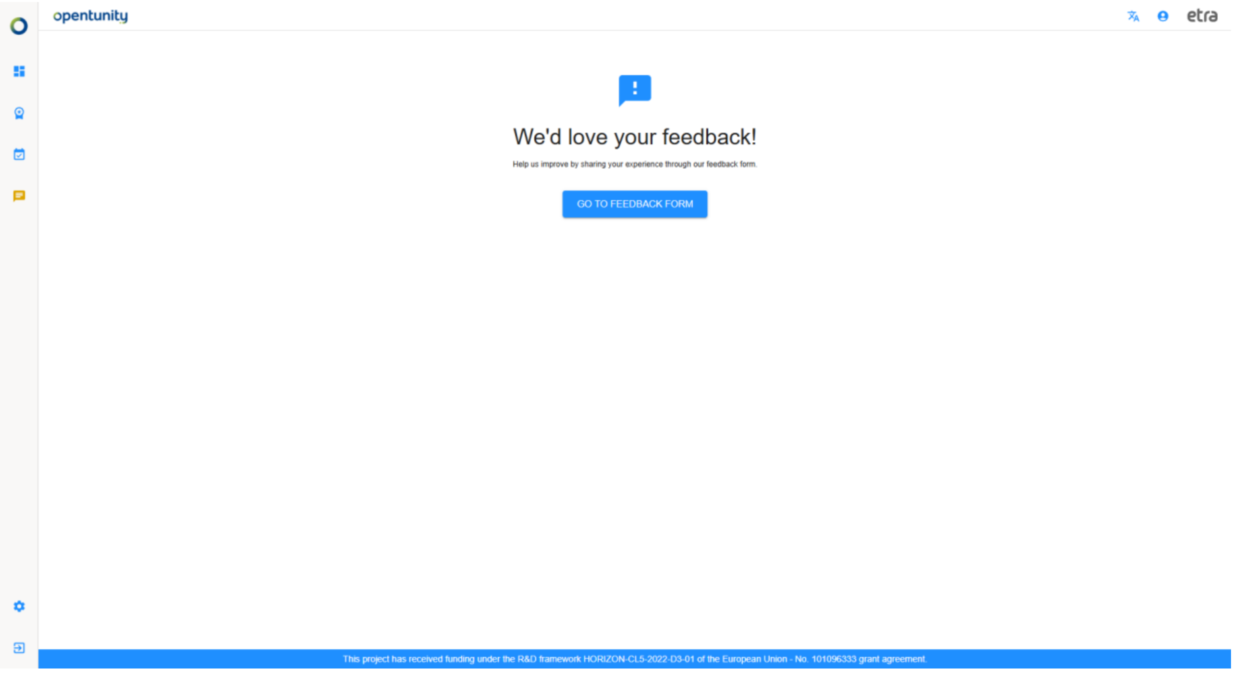


Figure 63. NILM graphical user interface, feedback page.

Figure 64. NILM feedback survey.

4.7 Application Example

This tool requires initial setup, including user creation and linking identifiers with the corresponding house or houses smart meters. Once this is complete, credentials for the platform are generated, but initially, only total consumption data is displayed on the dashboard.

First, the user needs to select their household appliances in the settings menu, see Figure 65. This process can also be done during credential creation if the energy retailer provides the information.

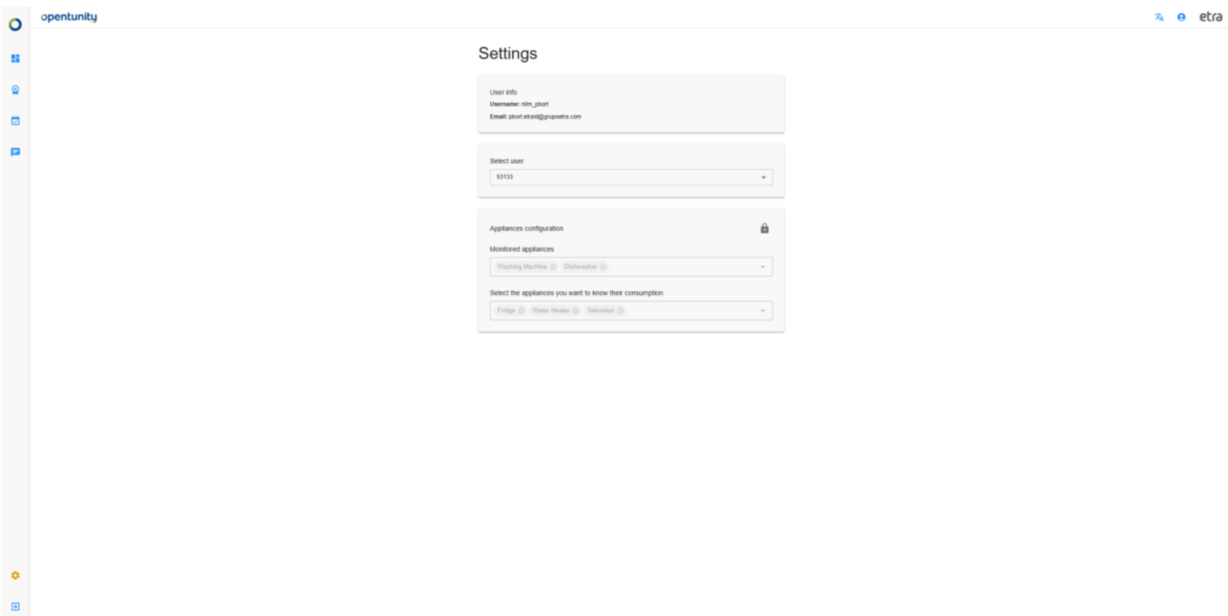


Figure 65: NILM graphical user interface, settings menu.

After listing all appliances, disaggregation begins, and the information appears on the dashboard, see Figure 59. The user can then navigate to the certificates page, see Figure 61, to create certificates by entering the requested information, see Figure 66. They can compare the theoretical value of the certificate with the actual one. Once a certificate is created, the information is stored and continuously updated. Appliance changes are considered, and the certificate can be recreated at any time.

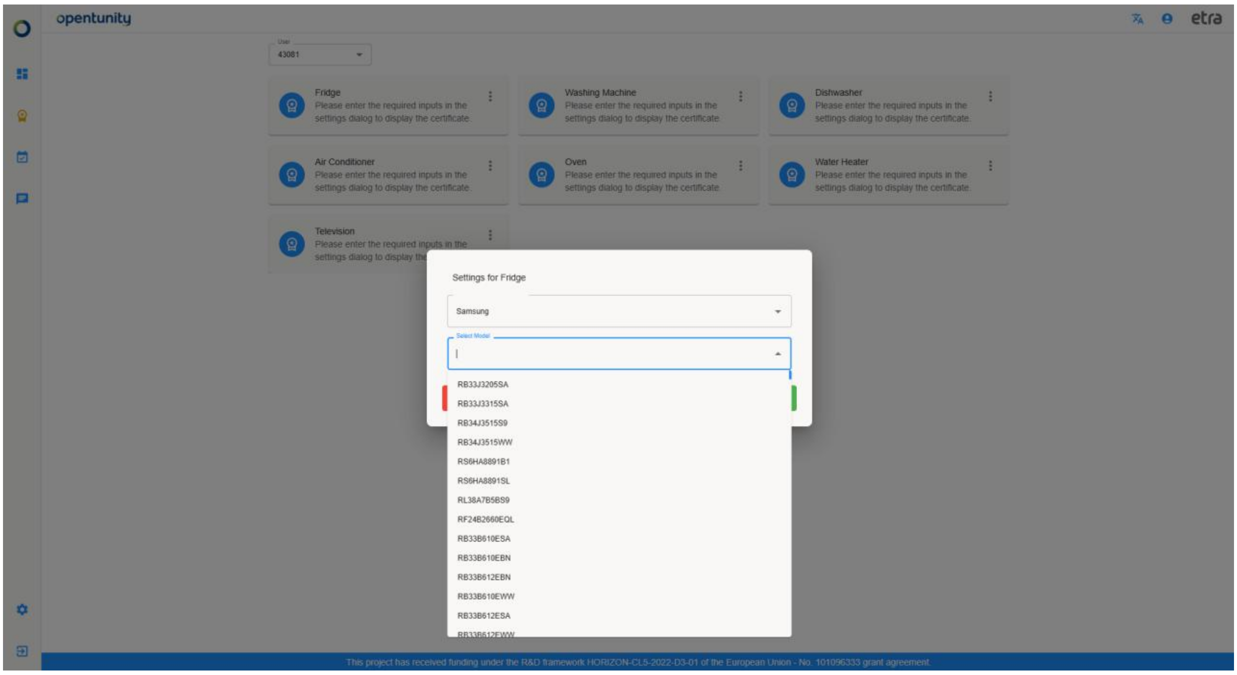


Figure 66: NILM graphical user interface, certificate settings.

4.8 Development & integration status

The development of the components is nearly complete, with everything functioning using data from different pilots. The main components that will undergo further refinement are the NILM GUI and the machine-learning models. Real data from the pilots will be used to refine and re-train the machine-learning models for improved inference performance. Their feedback will be used to further improve the Graphical User Interface.

4.9 Requirements in Equipment & Infrastructure

Intrusive Load Monitoring, as detailed in the methodology, is essential for evaluating and retraining machine-learning models. This process requires the installation of submeters within households to monitor appliances that contribute significantly to total energy consumption, such as dishwashers, washing machines, refrigerators, water heaters, and air conditioners. The main equipment requirements are:

- **Smart meters:** These devices monitor the total power consumption of the household with a minimum granularity of one minute, ensuring the recognition of typical appliance usage patterns.

- **Submeters:** These devices monitor the individual power consumption of selected appliances with the same granularity as smart meters, providing real data for retraining and evaluating machine-learning models.
- **Controllers:** No specific controllable devices are required for this task.

Regarding infrastructure, there are slight preferences for monitoring appliances that can be switched on or off at any time for flexibility market related use cases.

4.10 Assumptions and restrictions

The main restriction in applying NILM prediction components and the models developed so far is data granularity, as each appliance has a characteristic curve that defines it. If the granularity is insufficient, model performance will be compromised. Currently, the minimum granularity provided for the models is one minute, but increased granularity enhances AI model performance. As can be seen in Figure 67, the difference between both granularities lies in the amount of information given to the model explaining the consumption data, particularly visible at peaks. Therefore, the AI model is able to distinguish the appliances patterns.

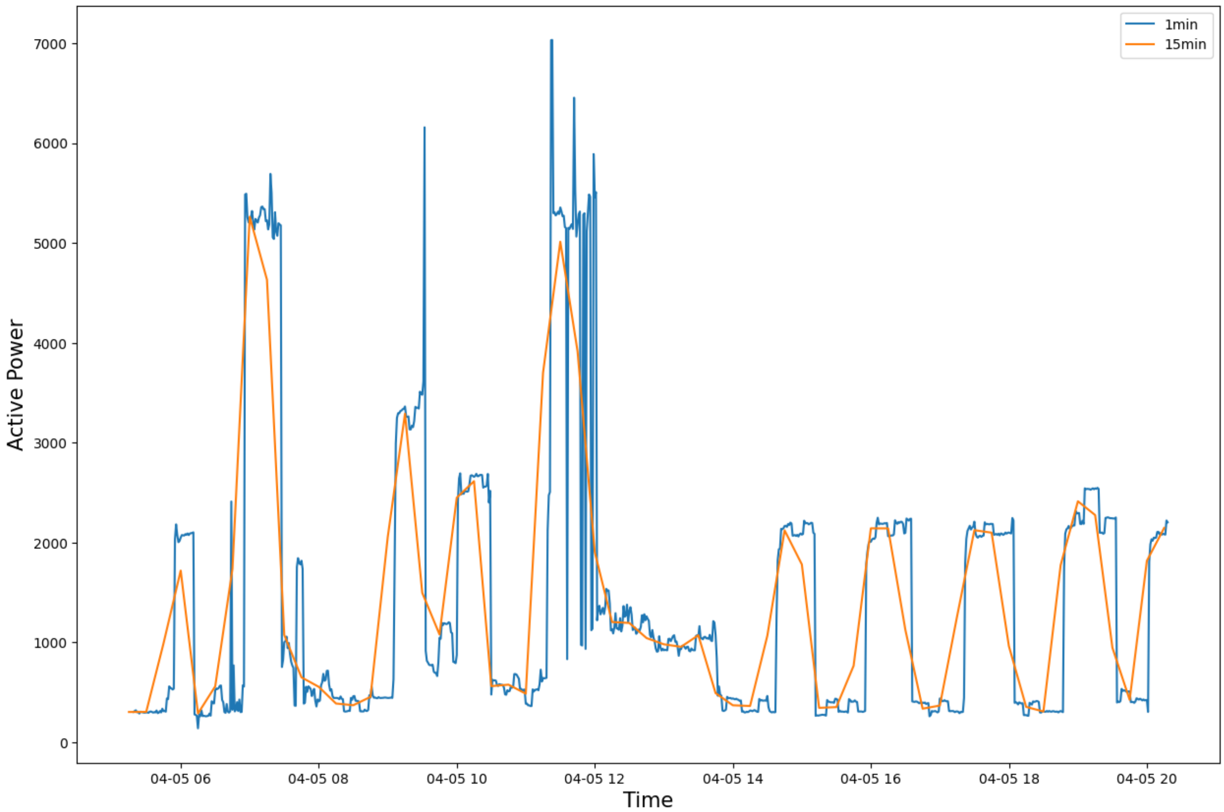


Figure 67. one minute vs fifteen-minute granularity example

4.11 Pre-validation activities & Preliminary tests

Validation activities and preliminary tests have been conducted, focusing on NILM machine-learning models to assess their predictive capabilities. Publicly available data from the internet and pilot data have been utilized for model validation and evaluation metrics creation. Main results can be seen in Table 15, Table 16, Table 17 and Table 18.

Several public datasets have been examined, but only those with large samples exceeding one month of data, high granularity with a minimum of one-minute intervals, and minimum active power in their exogenous variables were selected. The chosen datasets include UK-DALE [13], the Almanac of Minutely Power dataset (AMP) [14], Electricity Consumption and Occupancy (ECO) [15], REFIT [16], ENERTALK [17], and DEDDIAG [18].

4.11.1 Data Processing & Training Methods

Datasets have undergone a series of steps to ensure clean and clear information:

- **Resampling:** The datasets vary in granularity. Once all variables are ordered in the same pandas DataFrame, resampling functions are used to match the target granularity, which is one minute. The entire dataset is then divided into as many subsets as there are appliances and households.
- **Data pre-preprocessing and cleaning:** For each dataset, records with NaN and None values are removed. This process is applied to exogenous variables, primarily active and reactive power. The target variable, which is the appliance's individual active power, must also match these records and consider its own NaN and None values. An example is shown in Figure 68, using the ECO dataset.

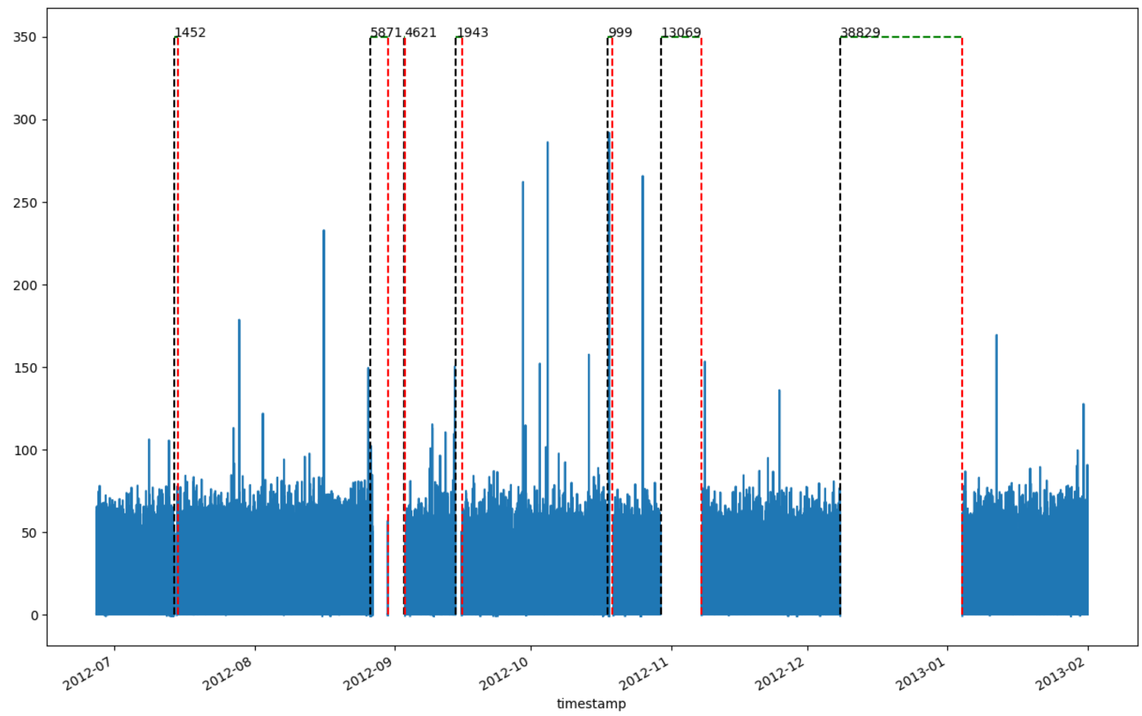


Figure 68: Example of ECO dataset NaN cleaning.

- **Feature extraction:** New variables are created to provide more immediate information to the models. Seasonal time variables such as day, hour, minute, weekday, month, and season are generated. Additionally, three hours of lags are created for each record to provide the model with past values of total active power and recognize typical curves.

- **Train and test dataset creation:** At this stage, a large dataset for each appliance is created by collecting all the relevant household datasets. Each dataset is divided into 90% training and 10% testing. The years of data per appliance for datasets containing only active power and those with both active and reactive power are shown in Figure 69 and Figure 70.

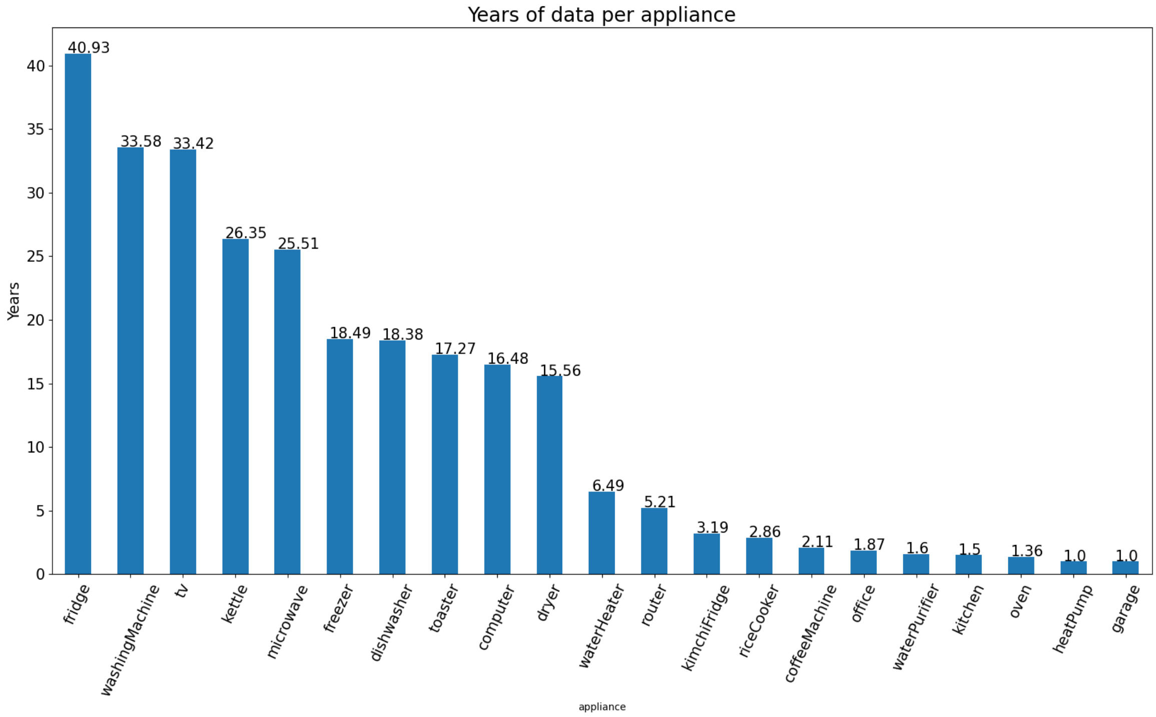


Figure 69: Years of data per appliance with only active power datasets

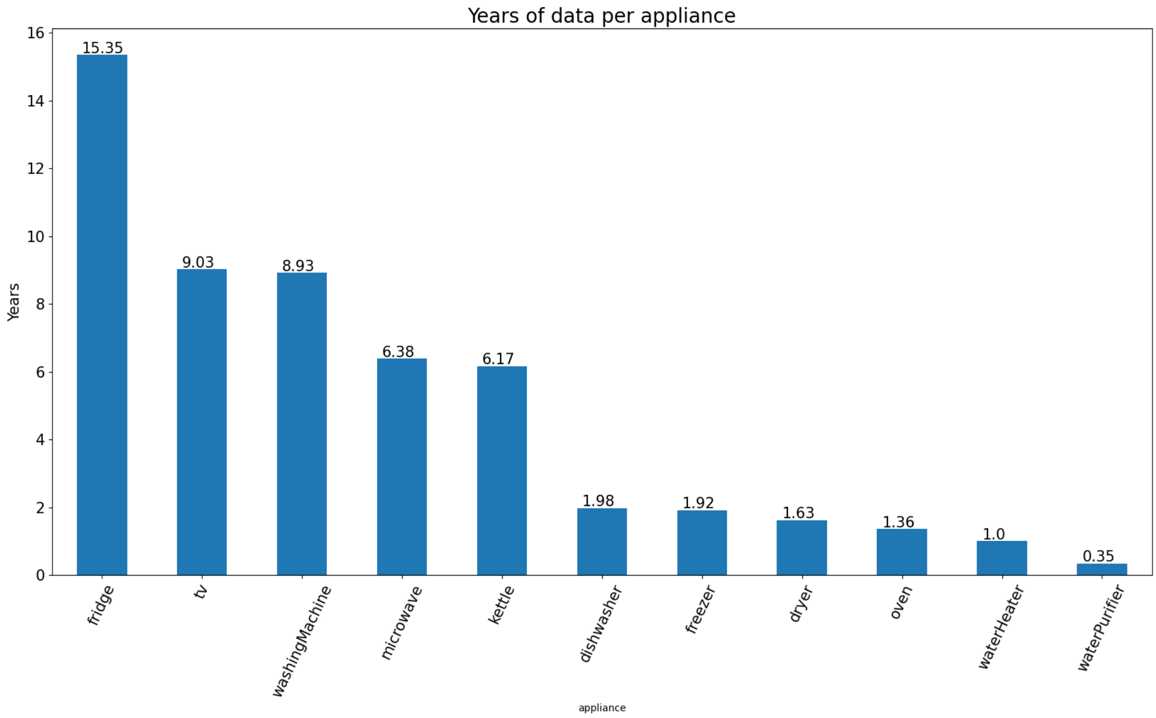


Figure 70: Years of data per appliance with active and reactive power datasets

Training methods can be divided into the following steps:

- **Data normalization:** To reduce complex model training calculations, datasets are normalized and scaled using the Standard Scaler:

$$z = \frac{(x - u)}{s}$$

where *x* is the value, *u* is the mean of the column, and *s* is the standard deviation. Other scalers like MinMax Scaler were also tested, but performance did not change significantly.
- **Model training:** Using the scaled training dataset, machine learning models are created with Cross Validators, specifically BayesianSearchCV and RandomizedSearchCV. Cross Validation involves dividing the training dataset into several folds, training the model on some folds, and testing on others. The model with the best evaluation metric on each fold is considered to generalize best to different data. Grid Search methodologies are used to find the best-performing model hyperparameters. The desired intervals for hyperparameter tuning are selected, and optimizers find the best-evaluated hyperparameters through Random searches or more complex techniques like BayesianSearchCV. After sufficient training sessions, the best-performing parameters are identified. This process was tested with XGBoostRegressor models and TensorFlow neural network architectures.
- **Model evaluation:** The proposed hyperparameter search optimizers are evaluated on each fold using metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R^2. MAE was the primary metric used. Once the models are created, their performance is further tested against the test datasets, which are real users from the Spanish pilot, using classification metrics. Specifically, the F1-score is employed to evaluate the accuracy of appliance activation (ON/OFF) detections, rather than just predicting continuous power values. This evaluation is performed by converting both the predicted and real power sequences into binary activation vectors. For each time step, a threshold is applied to determine if the appliance is considered "ON", typically set to a small positive power value to avoid misclassifying slight fluctuations or predictive noise as false positives. The same threshold is applied to both the real power and the predicted output. Each time point is then classified as a true positive, false positive, true negative, or false negative. The resulting F1-score balances the precision and recall values to provide a robust evaluation of event detection performance. Yielding the following results:

Table 15: NILM best models evaluation metrics results for each appliance using active power only and one minute granularity

Appliance	F1-score (%)
Washing Machine	59.13
Dishwasher	61.49
Fridge	93.15
Oven	73.68
Air Conditioner	88.15

Table 16. NILM best models' evaluation metrics results for each appliance using active and reactive power and one minute granularity.

Appliance	F1-score (%)
Washing Machine	59.07
Dishwasher	57.68
Fridge	84.75
Oven	66.05
Air Conditioner	96.57

Table 17: NILM best models evaluation metrics results for each appliance using active power only and fifteen-minute granularity

Appliance	F1-score (%)
Washing Machine	0.08
Dishwasher	55.56
Fridge	91.81
Oven	42.62
Air Conditioner	81.66

Table 18. NILM best models' evaluation metrics results for each appliance using active and reactive power and fifteen-minute minute granularity.

Appliance	F1-score (%)
Washing Machine	20.00
Dishwasher	78.50
Fridge	96.14
Oven	60.61
Air Conditioner	88.59

- **Model uploading:** The final step is to upload the trained and evaluated models to the Model serving platform MLflow, as shown in Figure 55. The metrics and exogenous variables are also registered with the model.

4.11.2 Tests & Results

Model testing and evaluation metrics are presented in Table 15, Table 16, Table 17 and Table 18. The main results indicate that reactive power is a useful variable for determining appliance behaviour, showing superior overall results. However, datasets containing both variables are limited, as shown in Figure 69 and Figure 70.

Preliminary conclusions suggest the necessity of high granularity and exogenous variables to incrementally enhance the model's performance. Generalization remains challenging in machine learning projects, requiring substantial amounts of detailed training and testing data on appliances. One potential opportunity for improving model performance is to re-train generalized models with specific household appliance electrical measurements, providing them occupant needs and behaviours. An example can be seen with the air conditioner model performance over the Spanish pilot, since the training data used is from the same households as the testing data, the predictions are very accurate. Alternatively, training specific machine-learning models for individual households during the sub-metering process, followed by continued monitoring without sub-metering using NILM, could be explored.

The results obtained so far in appliance disaggregation using NILM techniques are promising, although preliminary, given the limited amount of data collected from recent pilot deployments. The models have shown notable performance, particularly in the case of the air conditioner, where training with data from the same household as the test set led to high accuracy, achieving an F1-score of up to 96.57 %, which is expected under these conditions.

It is important to note that the primary evaluation metric used is the F1-score, which is more robust than the commonly reported accuracy (≈80 % in the state of the art). F1-score offers a better representation of ON/OFF event detection performance by balancing precision and recall. In our case, F1-scores range from 20 % to 96 %, depending on the appliance and data granularity, highlighting both the strengths and current limitations in terms of model generalization.

A key area for future improvement is the integration of user feedback via the Events page, see Figure 62, where users can validate or correct appliance detection events. This user interaction enables the refinement of existing models through retraining or fine-tuning based on household-specific usage patterns, ultimately improving long-term prediction accuracy.

In summary, these results represent a solid and scalable foundation for NILM deployment in real-world settings. Further improvements will require longer monitoring periods, greater household variability, and continued user engagement to unlock the full potential of the approach.

5. CONCLUSIONS

The deliverable presented the development of BFMS and NILM components, as envisaged in Tasks 4.1-4.3 of the OPENTUNITY project. The two tools mark a significant advancement in the identification of the usage patterns for individual loads and energy flexibility calculation and dispatch.

At first, the HEMS/BEMS located at the pilot sites of the project were described as the indispensable tools that provide the necessary data for BFMS and NILM. The HEMS/BEMS are responsible for the monitoring and control of flexibility assets, the streaming of sensorial data and the overall recording of the energy consumption of a dwelling.

BFMS provides a robust mechanism for unlocking flexibility potential, enabling market interactions and facilitating energy monetization. Based on power consumption data, it calculates day-ahead baseline and flexibility power consumption timeseries. The inclusion of DR Initialization Service facilitates DR participation for occupants, reducing barriers to engagement while ensuring that user comfort and constraints are respected.

In parallel, the NILM component brings state-of-the-art load identification and disaggregation capabilities to the OPENTUNITY project. By leveraging advanced AI-driven methodologies, NILM is able to provide precise energy usage insights that can be used to feed BFMS, but also to identify usage patterns and optimize energy consumption.

Both tools have been tested and demonstrated using either available data from the project (BFMS), or publicly available dataset (NILM). The tools are in a fully operational mode for the next phase of the project, i.e., the demonstration activities at the pilot sites.

6. References and acronyms

6.1 References

[1] "ACCEPT Project," [Online]. Available: https://www.accept-project.eu/.

[2] Y. Sun, J. Li, J. Liu, B. Sun and C. Chow, "An improvement of symbolic aggregate approximation distance measure for time series," *Neurocomputing,* 2014.

[3] "ACCEPT D4.6 Citizen Digital Twin v2".

[4] "ACCEPT D4.8 Building Digital Twin v2".

[5] M. Ester, H. P. Kriegel, J. Sander and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Second International Conference on Knowledge Discovery in Databases and Data Mining*, 1996.

[6] D. Arthur and S. Vassilvitski, "K-means++: The advantages of a careful seeding," in *18th annual ACM-SIAM symposium on discrete algorithms*, 2007.

[7] C. Yuan and H. Yang, "Research on K-Value Selection Method of K-Means Clustering Algorithm," *Multidisciplinary Scientific Journal,* 2019.

[8] T. John, I. Sarantakos and T. T. Teo, "Stacking different services of an energy storage system in a grid-connected microgrid," *Renewable Energy,* 2022.

[9] N. Zografou-Barredo, C. Patsios, I. Sarantakos, P. Davison, S. Walker and P. Taylor, "MicroGrid Resilience-Oriented Scheduling: A Robust MISOCP Model," *IEEE Transactions on Smart Grid,* 2021.

[10] T. Sayfutdinov, C. Patsios, D. Greenwood, M. Peker and I. Sarantakos, "Optimization-based modelling and game-theoretic framework for techno-economic analysis of demand-side flexibility: A real case study," *Applied Energy,* 2022.

[11] Y. Yu, Y. Zhu, D. Wan, H. Liu and Q. Zhao, "A novel symbolic aggregate approximation for time series," *Advances in Intelligent Systems and Computing,* 2019.

[12] Y. Zhang, L. Duan and M. Duan, "A new feature extraction approach using improved symbolic aggregate approximation for machinery intelligent diagnosis," *Measurement: Journal of the International Measurement Confederation,* 2019.

[13] "ISO 13790 Energy performance of buildings - Calculation of energy use for space heating and cooling," 2008.

[14] D. Crawley, L. Lawrie, F. Winkelmann, W. Buhl, Y. Huang, C. Pedersen, R. Strand, R. Liesen, D. Fisher, M. Witte and J. Glazer, "EnergyPlus: creating a new-generation building energy simulation program," *Energy and Buildings,* 2001.

[15] S. Klein, W. Beckman and J. Duffie, "TRANSYS - A Transient Simulation Program," *ASHRAE Transactions*, 1976.

[16] K. Sinaga and M. Yang, "Unsupervised K-Means Clustering Algorithm," *IEEE Access*, 2020.

6.2 Acronyms

Table 19. Acronyms

Acronym	Explanation
AC	Air condition
API	Automatic Programming Interface
BEMS	Building Energy Management System
BFMS	Building Flexibility Management System
DHW	Domestic Hot water
DR	Demand Response
ESS	Energy storage system
EV	Electric Vehicle
EU	European Union
EWH	Electric Water Heater
FSP	Flexibility Service Provider
GP	Gaussian Processes
GUI	Graphical User Interface
HEMS	Home Energy Management System
HVAC	Heating Ventilation and Air Conditioning
IoT	Internet of Things
LV	Low Voltage
NILM	Non-Intrusive Load Monitoring
OCPP	Open Charge Point Protocol
PV	Photovoltaic
PODs	Point of Deliveries

R&D	Research and Development
RF	Random Forest (RF)
SAX	Symbolic Aggregate Approximation
SO	System Operator
UI	User Interface
UC	Use Case
WP	Work Package
WSN	Wireless Sensor Network