D6.1 – Adaptive Building Operation and Management System (BOMS) v1

This project has received funding from the European Union's Horizon Europe research and innovation programme under the grant agreement number 101147722. The European Union is not liable for any use that may be made of the information contained in this document, which is merely representing the authors' view.

Project Acronym:	AccesS
Project Full Title:	Enhancing Accessibility and Sustainability in Smart Cities and Smart Buildings: The Universal Accessibility Suite Initiative
Grant Agreement:	101147722
Project Duration:	36 months (01/06/2024 – 31/05/2027)

DELIVERABLE 6.1 Adaptive Building Operation and Management System (BOMS) v1

Work Package:	WP6 – Intelligent Systems for Adaptive Building and Mobility Management
Task:	T6.1 – Adaptive Building Operation and Management System (BOMS)
Document Status:	Final
File Name:	AccesS_D6.1_Adaptive Building Operation and Management System (BOMS) v1_CERTH
Due Date:	30.11.2025
Submission Date:	15.12.2025
Lead Beneficiary:	CERTH

Dissemination Level	
Public	☒
Sensitive	☐

Authors List

Leading Author(s)			
First Name	**Last Name**	**Beneficiary**	**Contact e-mail**
Ioannis	Koskinas	CERTH	jkosk@iti.gr
Napoleos	Bezas	CERTH	napobezas@gmail.com

Co-Author(s)				
#	**First Name**	**Last Name**	**Beneficiary**	**Contact e-mail**
1	Dimitris	Tzilopoulos	CERTH	dimtzilop@iti.gr
2	Panagiota	Chatzipanagiotidou	CERTH	phatzip@iti.gr
3	Ariadni	Elmaloglou	QUE	a.elmaloglou@que-tech.com
4	Mija	Susnik	DMO	Mija@demobv.nl
5	Fatemeh	Asgharzadeh	DMO	fatemeh@demobv.nl
6	Mohamed	Elagiry	DMO	Mohamed@demobv.nl
7	Ilias	Aliferis	UNIS	AliferisI@unisystems.gr
8	Raquel	Ortega Martínez	CETEM	r.ortega@cetem.es
9	Fabio	Garista	VAIMOO	fabio.garista@angel4future.com

Reviewers List

Reviewers			
First Name	**Last Name**	**Beneficiary**	**Contact e-mail**
Dimitris	Botsis	UNIS	botsisd@unisystems.gr
Stefano	Mauri	AEM	smauri@aemsa.ch

Version History

v	Author	Date	Brief Description
0.1	Napoleon Bezas, Ioannis Koskinas, CERTH	21.07.2025	Initial draft (ToC)
0.3	All authors, CERTH	10.09.2025	First version with initial content for all sections
0.4	Napoleon Bezas, Ioannis Koskinas, CERTH	25.11.2025	Ready for internal review
0.6	Napoleon Bezas, Ioannis Koskinas, CERTH	04.12.2025	Updated based on peer review
0.7	All authors, DMO	11.12.2025	Updates in Section 7 on GIS
1.0	Dimosthenis Ioannidis, Panagiota Chatzipanagiotidou, CERTH	15.12.2025	Final version ready for submission

Legal Disclaimer

The AccesS project has received funding from the European Union's Horizon Innovation Action under grant agreement No 101147722. The sole responsibility for the content of this publication lies with the authors. It does not necessarily reflect the opinion of the European Climate, Infrastructure and Environment Executive Agency (CINEA) or the European Commission (EC). CINEA or the EC are not responsible for any use that may be made of the information contained therein.

Copyright

Executive Summary

This deliverable presents the development of an intelligent decision support tool aimed at enhancing the real-time monitoring and adaptive optimization of building operations i.e., the Building Operation and Management System (BOMS), developed as part of the activities of T6.1 Adaptive Building Operation and Management System. In response to growing demands for energy efficiency, user comfort, and operational flexibility, the tool leverages cutting-edge technologies—such as smart sensors, Internet of Things (IoT) devices, and deep learning models—to transform how buildings respond to dynamic conditions and evolving user needs.

A central component of this work is the implementation of a robust, scalable database that serves as the backbone for data storage, access, and analysis. A comprehensive data model has been designed to structure and standardize data collected from diverse sources, particularly the pilot sites. All incoming data from these pilots—ranging from environmental measurements to energy metrics—are systematically parsed, validated, and stored. This ensures consistency, integrity, and readiness for further processing.

To facilitate secure and interoperable access to the stored data, a dedicated API has been developed. This interface enables project partners and external systems to query and retrieve data efficiently, supporting the integration of the tool within a broader ecosystem of digital services. The API also supports real-time data flows, allowing external applications to contribute to or react to system insights dynamically.

The system includes a powerful visualization component that provides intuitive graphs and dashboards covering a variety of measurements from all pilot sites. These visualizations enhance transparency and provide stakeholders with actionable insights into building performance, environmental conditions, and energy consumption patterns.

At its core, the tool continuously gathers and analyzes real-time data from building environments to support intelligent, adaptive decision-making. Through the integration of deep neural network algorithms, the tool significantly improves the forecasting of energy consumption and renewable energy production. These predictive insights enable smarter resource allocation and energy management, contributing to the sustainability and resilience of the built environment.

Beyond forecasting, the tool also includes a recommendation engine that suggests optimal operational strategies, ensuring adjustments remain within user comfort thresholds. The system is user-centric and location-aware, incorporating Geographic Information Systems (GIS) to include accessibility featured based on geographic and site-specific data.

A key innovation lies in the system's ability to operate synergistically with other digital tools of the AccesS Suite. Rather than functioning in isolation, it communicates with a wider network of data sources and services, enhancing its decision-making capabilities and ensuring responsiveness to both internal performance indicators and external contextual changes.

In summary, D6.1 introduces a powerful, adaptive, and modular platform that brings together advanced data processing, artificial intelligence, and user-focused design. It establishes a solid foundation for intelligent building management systems that are not only energy-efficient and cost-effective but also capable of evolving in line with user needs and technological advancements.

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

Term	Description
API	Application Programming Interface
BOMS	Building Operation Management System
CSV	Comma Separated Values
DSS	Decision Support System
GIS	Geographic Information Systems
HTTPS	Hyper Transfer Protocol
IoT	Internet of Things
ISO	International Organization for Standardization
JWT	Json Web Tokens
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MQTT	Message Queuing Telemetry Transport
RES	Renewable Energy Sources
REST	Representational State Transfer
SMAPE	Symmetric Mean Absolute Percentage Error
SQL	Structured Query Language
SSO	Single Sign-On
UTC	Universal Time Zone
WAPE	Weighted Absolute Percentage Error
WHO	World Health Organization

1 Introduction

1.1 Scope and objectives of the deliverable

This deliverable (D6.1) presents the development of the Building Operations Management System (BOMS), which supports the optimization of building operations within the AccesS Suite. As part of this work, a monitoring dashboard was developed to collect, visualize, and manage data from all pilot sites in real time. A TimescaleDB database was implemented to store and organize all pilot measurements, supported by a dedicated data model and an API that enable efficient data exchange between BOMS components and other systems within the AccesS Suite.

Building on this data infrastructure, two key sub-modules were developed: the Decision Support System (**DSS**) and the Forecasting Engine (**Chronos**[1]). The DSS uses intelligent agents to generate actionable recommendations for building operations, helping to improve energy efficiency, maintain occupant comfort, and enhance system flexibility. The Forecasting Engine, based on Chronos models, employs AI methods to predict energy consumption, renewable energy production, and indoor conditions, providing essential inputs to the DSS. Together, these modules support adaptive, data-driven, and intelligent management of building operations. In addition, the RE GIS module supports the AccesS BOMS by enabling spatially explicit evaluation of building accessibility features.

This deliverable aligns with Task 6.1, led by CERTH in collaboration with QUE, DMO, UNIS, CETEM, and VAIMOO, focusing on data-driven and adaptive approaches to sustainable building management.

1.2 Structure of the deliverable

This report is structured as follows:

- **Section 2** describes the database design for storing all pilot measurements.
- **Section 3** showcases the process of pilot data parsing and ingestion to the database.
- **Section 4** focuses on the forecasting engine for energy consumption and RES production.
- **Section 5** describes the decision support tool designed to support operational decisions.
- **Section 6** presents the analytics and visualization components.
- **Section 7** presents the RE GIS module enabling spatially explicit evaluation of building accessibility.
- **Section 8** concludes summary and next steps.
- **Section 9** lists the references and source materials used.

Each section builds upon the previous, ensuring a logical flow from system design to application, and laying the foundation for future enhancements.

1.3 Relation to Other Tasks and Deliverables

The BOMS architecture was first defined in Task T1.3 *AccesS Architecture, System Requirements, and Integration*, ensuring alignment with the overall system design and integration framework established in WP1. Building on this foundation, Task T6.1 focuses on developing the BOMS system, coupled also with GIS, as a core component of the intelligent building management tools within WP6.

[1] https://github.com/amazon-science/chronos-forecasting

BOMS and GIS are closely related to other work packages within the AccesS project. In WP2, user needs and accessibility principles guide the design of user-centric features and ensure that the system supports inclusive building management. Within WP7, the system is integrated into the AccesS Suite, providing a unified dashboard for monitoring pilot metrics and ensuring interoperability with other digital tools. The connection with WP9 and WP10-WP12 allows the system to be validated and demonstrated in real pilot environments, ensuring that its functionalities are effective, scalable, and responsive to real-world conditions.

As a result, D6.1 establishes the main functionalities of the BOMS system coupled with GIS, together with the database and API that support integration and data sharing with external tools.

2 Integrated Data Management: Database, Ontology and API Layers

This chapter presents the core components of the platform's integrated data management architecture, focusing on the database structure, the ontology framework, and the API layer that together enable consistent, structured, and accessible data flow across the system.

To support the needs of multiple pilot buildings, each providing different datasets related to indoor conditions and energy performance (consumption and generation), a centralized data infrastructure has been established. At the heart of this infrastructure lies a purpose-built ontology, designed to unify and standardize semantic concepts across the platform. The ontology facilitates interoperability between internal tools by providing a common understanding of data entities and their relationships.

Complementing the ontology is a robust database schema that stores curated, structured data in a way that reflects the semantic model, ensuring consistency and traceability. On top of this data layer, an application programming interface (API) has been developed to enable efficient and secure access to the data by various tools and services within the platform.

This chapter outlines the design principles, architecture, and implementation of the ontology, database, and API, highlighting their roles in ensuring interoperability, scalability, and extensibility of the platform's data management capabilities.

2.1 Database

To efficiently manage the large volume of time-dependent data collected from the pilot buildings as they presented in more detail in Table 9 in 3.2., such as indoor environmental conditions and energy consumption/generation metrics, the platform utilizes PostgreSQL with the TimescaleDB extension. TimescaleDB extends PostgreSQL with native support for time-series data, offering optimized storage, fast querying, and scalability for datasets that grow continuously over time. This choice enables the platform to handle high-ingestion rates and perform complex temporal queries while maintaining compatibility with standard SQL and the broader PostgreSQL ecosystem. The database schema is aligned with the platform's ontology, ensuring semantic consistency between the stored data and the concepts used across the system.

In the sections that follow, we describe the data model, the organization of measurements and metadata collected.

2.1.1 Database Model Overview

The platform's database schema is designed to manage environmental and energy-related data across multiple monitored buildings. It supports sensor integration, alarm logic, time-series storage, and forecast modeling. The schema is normalized, relational, and enhanced with TimescaleDB for handling high-volume time-series data efficiently. Core entities:

- Buildings & Spaces: Represent the physical hierarchy.
- Alarms & Notifications: Represent the rule-based logic for alerts.
- Sensors Data: Represent the measurement sources and values.
- Forecast Data: Represent modeled/predicted data.

2.1.1.1 Buildings & Spaces

The buildings table stores information about the monitored physical structures within the platform. Each building acts as the top-level entity in the spatial hierarchy and is linked to multiple spaces. This table serves as the entry point for associating environmental data with physical locations

Table 1. Example of buildings in the database

Column	Type	Description	Constraints
id	text	Unique building ID	Primary Key
name	text	Optional building name	Nullable

The spaces table represents individual rooms, zones, or sub-areas within a building. Each space is associated with a single building and may contain multiple sensors. This table is essential for organizing data spatially and enables room-level analysis and monitoring.

Table 2. Example of spaces in the database

Column	Type	Description	Constraints
id	text	Unique space ID	Primary Key
name	text	Optional space name	Nullable
building_id	text	Parent building ID	Foreign Key

2.1.1.2 Alarms & Notifications

The alarms table defines threshold-based monitoring rules tied to specific sensors, spaces, and buildings. When a metric surpasses a defined condition (e.g., temperature > 28°C), the alarm logic is triggered. This model encapsulates the rule logic, message, and configuration needed for real-time alerting.

Table 3. Example of alarms in the database

Column	Type	Description	Constraints
id	uuid	Unique alarm ID	Primary Key
sensor_id	text	Associated sensor ID	Foreign Key
space_id	text	Related space ID	Foreign Key
building_id	text	Related building ID	Foreign Key
metric_type	text	Metric monitored (e.g., temp)	Not Null
title	text	Alarm title	Not Null
message	text	Alarm message	Not Null
operator	text	Comparison operator	Not Null
operand	float	Threshold value	Not Null

The alarm notifications table logs instances when alarms are triggered. Each notification references a specific alarm and includes a seen flag to track whether it has been acknowledged. This table enables user-facing alert management and historical tracking of abnormal conditions.

Table 4. Example of alarm notifications in the database

Column	Type	Description	Constraints	Column
id	uuid	Notification ID	Primary Key	id
seen	boolean	Seen status	Default: false	seen
alarm_id	uuid	Linked alarm ID	Foreign Key	alarm_id

2.1.1.3 Sensors Data

The sensors data table stores raw time-series measurements from sensors deployed across buildings and spaces. These include metrics such as temperature, humidity, or energy consumption. The table is designed as a TimescaleDB hypertable to support high-throughput ingestion and efficient querying of time-dependent data.

Table 5. Example of sensor data in the database

Column	Type	Description	Constraints
metric_type	text	Type of measurement	Not Null
units	text	Measurement units	—
metric_value	jsonb	Actual value(s) (flexible format)	—
building_id	text	Source building ID	Foreign Key
space_id	text	Optional space ID	Nullable
sensor_id	text	Optional sensor ID	Nullable
value_timestamp	timestamptz	Time of measurement	Not Null
sensor_type	text	Sensor category/type	Nullable

2.1.1.4 Forecast Data

The forecast data table stores predicted values generated by machine learning or statistical forecasting models. These forecasts mirror the structure of IoT data but are used to support proactive decision-making. The table tracks the forecasted metric, prediction timestamp, source model, and associated location context.

Table 6. Example of forecast data in the database

Column	Type	Description	Constraints
metric_type	text	Type of forecasted metric	Not Null
metric_value	jsonb	Forecast value (e.g., time-value map)	—

model_id	text	ID of prediction model used	Not Null
building_id	text	Associated building	Foreign Key
space_id	text	Optional associated space	Nullable
sensor_id	text	Optional sensor ID	Nullable
value_timestamp	timestamptz	Forecasted time	Not Null
sensor_type	text	Type/category of sensor	Nullable
units	text	Measurement units	—

2.1.1.5 Relations

The database schema is designed with relational integrity in mind, reflecting the logical connections between core entities such as buildings, spaces, sensors, alarms, and time-series data. These relationships ensure that:

Data is properly contextualized (e.g., sensor data is tied to specific spaces and buildings),
Redundancy is minimized through normalization,
Queries across entities are efficient and semantically meaningful.

Each model defines foreign key relationships using object or array relations:

- One-to-One / Many-to-One ("object"): A record belongs to a single related entity.
- One-to-Many ("array"): A record has multiple related children.

Composite Relations: Some relations (e.g., alarms to sensors) are based on multiple fields (e.g., sensor_id, space_id, and building_id) to preserve uniqueness and traceability across hierarchical structures. The following sections describe the specific relationships defined for each table and how they contribute to the semantic and operational cohesion of the data model.

Table 7. Example of entity relationships in the database

Entity	Related To	Type
buildings	spaces	1:N
spaces	sensors, iot_data	1:N
alarms	sensors, spaces, buildings	N:1
alarm_notifications	alarms	N:1
Sensor_data	buildings, spaces	N:1
forecast_data	buildings, spaces	N:1

2.2 API

The platform provides a RESTful API designed for efficient and secure access to structured data, including time-series measurements (e.g., indoor conditions, energy consumption, and generation) and related metadata. The API serves as the main interface for both internal tools and external integrations, ensuring semantic consistency across all platform components. All endpoints follow standard REST conventions and are versioned under /api/v1. Communication is handled through JSON-formatted requests and responses.

From a technical perspective, the API is built on a microservice-based architecture, developed using Node.js and more specifically Express framework[2], ensuring scalability, performance, and maintainability. Each service exposes a well-defined interface aligned with the platform's ontology and data model. All communication is secured via HTTPS, and JWT-based authentication combined with role-based access control ensures granular authorization. To optimize performance and reliability, the API supports pagination, query filtering, and data aggregation for large time-series datasets. Common operations are cached to reduce response latency, while rate limiting mechanisms protect against excessive or abusive requests. Consistent error handling follows standard HTTP response codes with descriptive JSON error messages.

The API is organized into several functional modules, including:

- **User Management & Authentication** – Manage users, roles, permissions, and authentication tokens.
- **Buildings & IoT Devices** – Access, configure, and monitor buildings, spaces, and sensors.
- **Measurements & Forecasting** – Retrieve real-time and forecasted environmental and energy data.
- **Alarms & Notifications** – Manage alerts, alarm conditions, and related event notifications.
- **Tools, Services & Files** – Register tools, manage uploaded files, and access service metadata.
- **Reporting & Automation** – Generate forecasting reports, manage scheduled tasks, and trigger platform actions.

A dedicated Swagger interface provides interactive documentation, allowing users to explore all endpoints, view parameters, and test requests directly within the browser. A screenshot of the Swagger is provided in Figure 1:

[2] Express framework: https://expressjs.com/

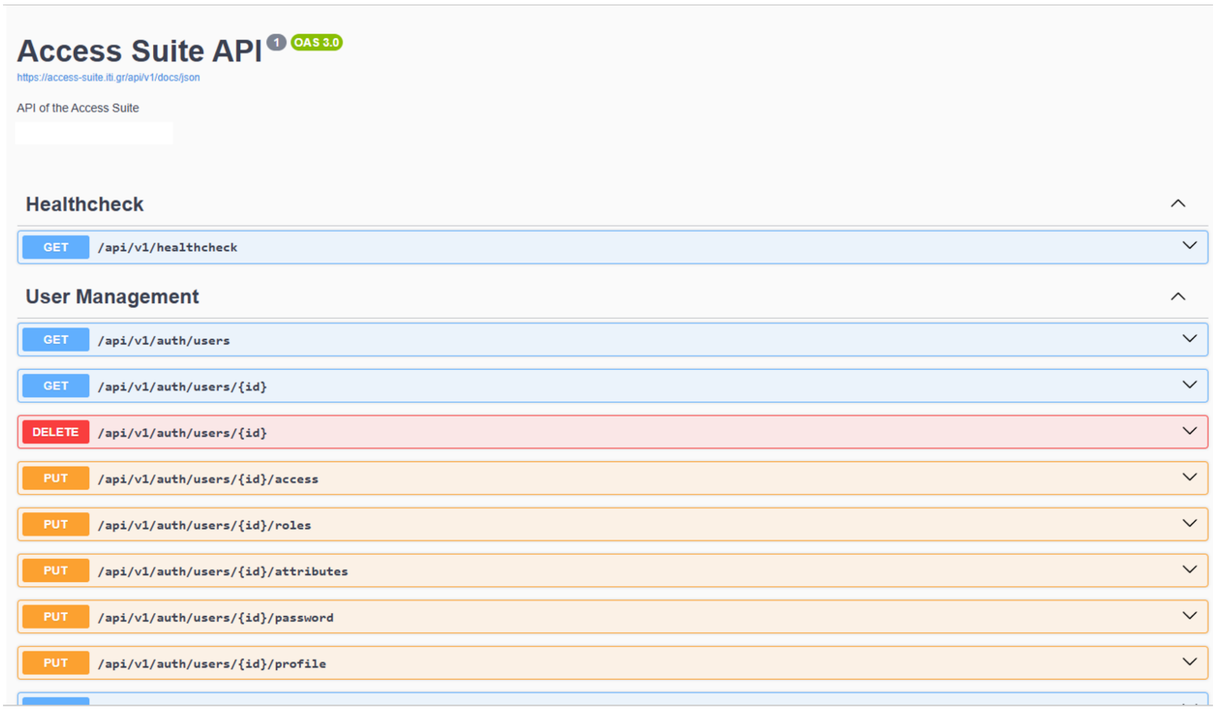

Figure 1. AccesS Suite - Swagger Interface

2.3 Ontology

A core element of the platform's architecture is its custom-designed ontology, which defines the communication schema between internal components. The ontology provides a structured and consistent semantic layer that describes key concepts and relationships relevant to the platform's domain, including indoor environmental data, energy usage, generation metrics, and the spatial organization of buildings. This unified model enables different tools within the platform to interpret and exchange data in a coherent way, regardless of their internal data formats or processing logic. By abstracting and formalizing the meaning of entities such as rooms, sensors, devices, and measurements, the ontology ensures that data remains context-aware, traceable, and interoperable across the platform.

In this section, we outline the design principles of the ontology, the main entities and their interrelations, and how the model supports integration, data retrieval, and interaction through the API layer.

2.3.1 Schema

The communication schema defines how core entities like buildings, spaces, sensors, and measurements are structured and linked within the platform. It translates the ontology into a practical data model, ensuring consistent and meaningful data exchange across components. By enforcing a shared structure, the schema enables tools to interpret and use data uniformly, supporting seamless integration, interaction, and retrieval.

2.3.1.1 Entities

Entities represent real-world objects or concepts, defined with a unique identifier and a set of properties.

- Sensor: Represents a physical device measuring environmental or power-related data.
- Building: Represents a physical structure containing spaces and sensors.
- Measurement: Represents a data point collected by a sensor at a specific time.
- Alarm: Represents a triggered event when sensor data meets certain conditions
- Alarm Notification: Represents notifications generated from alarms for users or system

2.3.1.2 Properties

Properties describe attributes of entities and their data types.

- ID (string): Unique identifier for entities like sensors or alarms.
- Timestamp (string, date-time format): The exact time when a measurement was taken.
- Metric Type (string): Type of measurement, e.g., temperature, humidity, power.
- Building ID (string): Reference to the building entity containing the sensor.
- Sensor Type (string, enum): Categories such as indoor_conditions or power_measurements.
- Value (number or string): The measurement value.

2.3.2 Indoor Conditions Schema

This API provides access to IoT sensor measurement data, enabling you to query time-series data for sensors within buildings or spaces. You can filter results by building, space, or sensor IDs, specify types (e.g., **indoor_conditions** or **power_measurements**), and define a time range using **start_date** and **end_date** (maximum range: 1 month).

Key features include:

- Filtering by multiple IDs (building, space, sensor)
- Sorting results (**order_by**)
- Grouping data points (**grouped**)
- Performing aggregations such as average, max, min, or sum over custom intervals (e.g., every 10 minutes)
- Retrieving metadata for columns, including sensor types and units
- Fetching first or last values for a specific metric or sensor.

Example use case:

This example demonstrates fetching time-series measurement data for sensors installed in a specific building. By specifying query parameters such as building ID, sensor type, date range, and sorting order, clients can retrieve detailed sensor readings, including metrics like temperature or humidity, with precise timestamps. This enables efficient monitoring and analysis of environmental conditions within the building.

Example Output:


```
{
  "data": [
    {
      "sensor_id": "sensorA1",
      "metric_type": "temperature",
      "value": 22.5,
      "unit": "C",
      "value_timestamp": "2025-07-01T01:00:00Z"
    },
    {
      "sensor_id": "sensorA1",
      "metric_type": "temperature",
      "value": 22.7,
      "unit": "C",
      "value_timestamp": "2025-07-01T02:00:00Z"
    }
    // Additional timestamped entries omitted for brevity
  ],
```

```
  "metadata": {
    "building_id": "building123",
    "sensor_type": "indoor_conditions",
    "record_count": 168,
    "time_range": {
      "start": "2025-07-01T00:00:00Z",
      "end": "2025-07-07T23:59:59Z"
    }
  }
}
```

2.3.2.1 Buildings and Spaces

The API provides several endpoints to manage and retrieve information about buildings and their associated spaces and sensors. You can retrieve all spaces within a specific building by using the GET method on the /api/v1/pilots/buildings/{building_id}/spaces endpoint, where you specify the building's unique identifier in the path. This request returns a list of spaces that belong to the building.

Example Output:

```
{
  "data": [
    {
      "space_id": "space101",
      "name": "Lobby",
      "floor": 1,
      "area_sqm": 50,
      "usage": "Reception"
    },
    {
      "space_id": "space102",
      "name": "Office 1",
      "floor": 2,
      "area_sqm": 30,
      "usage": "Workspace"
    }
  ],
  "metadata": {
    "building_id": "building123",
    "count": 2
```

```
  }
```

You can get detailed information about a specific space within a building by making a GET request to /api/v1/pilots/buildings/{building_id}/spaces/{space_id}, where both the building ID and space ID are provided in the path. Similarly, updating a space's information is done by sending a PUT request to the same endpoint with the updated space data in the request body. The updated space entity is returned in response.

Example Output:


```
{
  "space_id": "space102",
  "name": "Office 1",
  "floor": 2,
  "area_sqm": 30,
  "usage": "Workspace"
}
```


If you need to remove a space from a building, you send a DELETE request to /api/v1/pilots/buildings/{building_id}/spaces/{space_id} with the relevant building and space identifiers. The API responds with information about the deleted space.

Regarding sensors, you can retrieve all sensors associated with a particular space inside a building by sending a GET request to /api/v1/pilots/buildings/{building_id}/spaces/{space_id}/sensors. To register a new sensor within that space, you use the POST method on the same endpoint, including the sensor details in the request body.

Example output:


```
{
  "data": [
    {
      "sensor_id": "sensorA1",
      "type": "temperature",
      "model": "T1000",
      "status": "active",
      "installation_date": "2024-05-10T12:00:00Z"
    },
    {
      "sensor_id": "sensorB2",
      "type": "humidity",
      "model": "H2000",
      "status": "active",
      "installation_date": "2024-06-15T08:30:00Z"
    }
  ],
  "metadata": {
```

```
    "building_id": "building123",
    "space_id": "space101",
    "count": 2
  }
}
```

For operations on a specific sensor, such as retrieving its details, updating its configuration, or deleting it, you use the endpoints /api/v1/pilots/buildings/{building_id}/spaces/sensors/{sensor_id} with GET, PUT, and DELETE methods respectively. All these operations require the building ID, space ID, and sensor ID to be specified in the path, and for updates, the new sensor data is included in the request body.

Across all these endpoints, optional headers such as authorization tokens and API keys may be required depending on the security settings.

3 Pilot Data Parsing and Ingestion

A modular and scalable framework has been developed to parse and ingest measurement data from multiple sources. These data streams typically include indoor environmental conditions, energy consumption, and on-site energy generation. The framework harmonizes heterogeneous inputs into a common format suitable for centralized processing, storage, and visualization.

3.1 Parsing Architecture

Each pilot site exposes its measurement data through its own technical setup. At this stage of the project, all participating pilots provide data via RESTful APIs, which serve as the primary mechanism for accessing environmental and energy-related measurements. While the current integrations rely on REST APIs, the system is designed to accommodate other formats in the future (e.g., CSV endpoints, MQTT streams, or batch uploads). To manage the differences in API structure, authentication methods, and data formats across pilots, a dedicated parser is implemented for each site. These parsers are responsible for retrieving, transforming, and harmonizing measurements into a unified format. Figure 2 provides an overview of the end-to-end data flow handled by a dedicated parser:

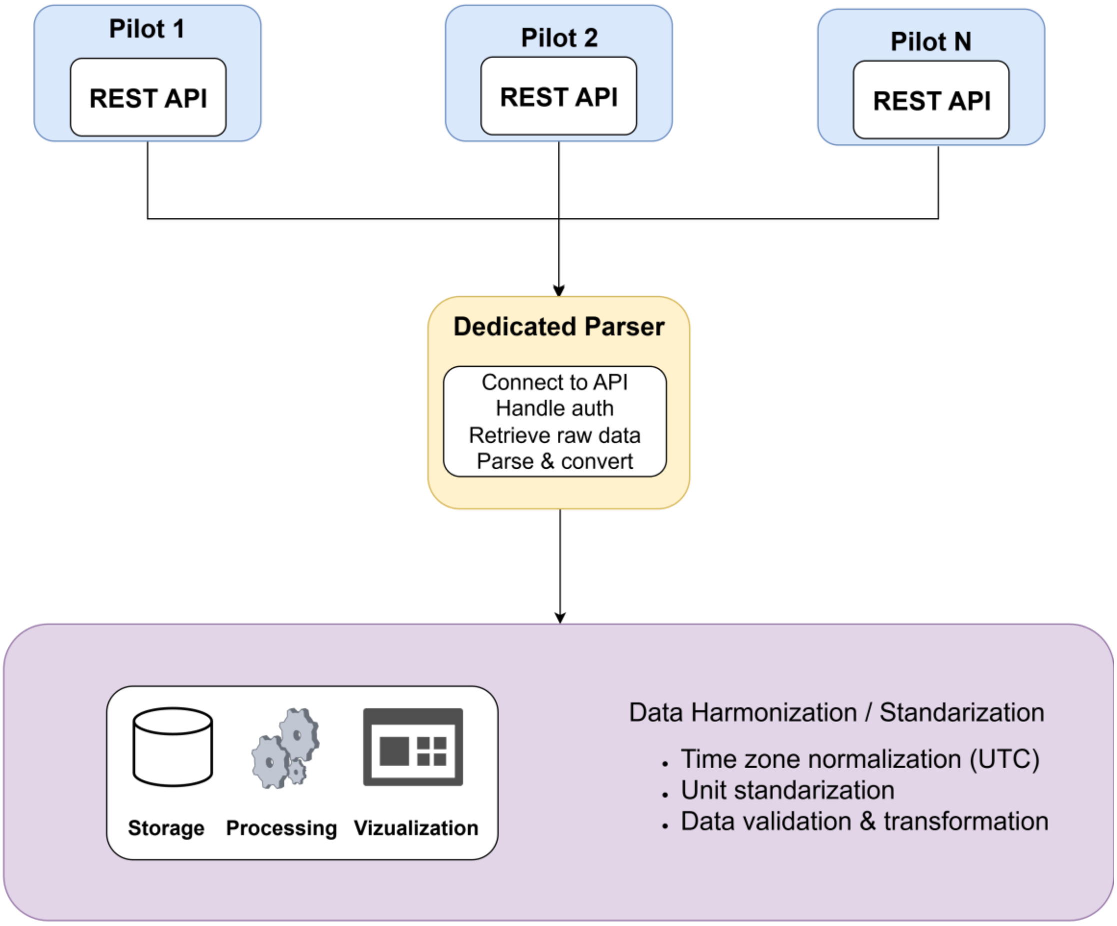

Figure 2. End-to-end data flow of the dedicated parser.

Each parser performs several key tasks to ensure consistency across data sources:

- Establishing connections to the pilot's API
- Handling authentication flows (e.g., token-based access)
- Retrieving raw measurement data
- Parsing and converting the data into a standardized structure

Each parser adheres to a universal interface defined by a shared base class, which enforces consistent logic across all integrations. This includes:

- API communication (with support for secure authentication)
- Time zone normalization, converting all timestamps to UTC
- Data Validation and transformation, including unit standardization
- Output formatting, aligned with a shared data schema

All parsed data is transformed into a harmonized schema (see Table 8), which ensures full interoperability across data sources and enables smooth integration into the platform's central storage, processing, and visualization components.

Table 8. Harmonized Data Schema for Pilot Measurements

Field Name	Description	Example
sensor_id	Unique identifier of the measuring device	id-01
space_id	Room or zone where the measurements was taken	room-01
sensor_type	Type of measurement	Indoor conditions, energy consumption/generation
metric_type	Type of measurement	C02, temperature
metric_value	Actual recorded value	780, 21.5
units	Standarized measurement units	ppm, celsius, kWh
value_timestamp	Timestamp of the measurement, normalized to UTC	2025-07-12T00:00:00+00:00
building_id	Identified of the building or pilot site	building_01

3.2 Multi-Source Integration

The framework is designed to integrate multiple data sources in a consistent and scalable manner. Each source is handled by a dedicated parser that transforms heterogeneous measurements into the shared harmonized schema. This modular approach allows new sources to be added efficiently without affecting existing data flows, ensuring interoperability across all integrated sites. Table 9 summarizes all measurement types currently monitored, their standard units, and the aggregation method used when combining data over time:

Table 9. Monitored Metrics.

Metric Type	Description	Units	Aggregation Method
consumption	Energy consumption	kW/kWh	Mean/Sum over time
production	Energy production	kW/kWh	Mean/Sum over time
co2	Carbon dioxide concentration	ppm	Average
voc	Volatile organic compounds	ppb	Average
no2	Nitrogen dioxide concentration	ppb	Average
rh	Relative Humidity	%	Average
temp	Temperature	°C	Average
pm1	Particulate matter (PM1)	µg/m³	Average
pm2.5	Particulate matter (PM2.5)	µg/m³	Average

Metric Type	Description	Units	Aggregation Method
pm10	Particulate matter (PM10)	µg/m³	Average

3.2 Scheduling and Centralized Ingestion

To automate and scale the ingestion process, a centralized scheduling system has been developed. For each configured pilot a background scheduler triggers data collection jobs at predefined intervals (e.g., every 5 min., every hour), depending on the capabilities and update frequency of the data source. Each scheduled job performs the following steps:

- Authentication & Fetching: The parser handles authentication (e.g., token retrieval) and requests the latest data from the pilot's API.
- Parsing & Normalization: Raw data is parsed and transformed into the shared schema.
- Storage: The parsed data is inserted into a centralized TimescaleDB database using a consistent schema across all pilots.
- Notification: A lightweight notification is sent to inform other components (such as the dashboard) of new data availability.

The scheduler is fault-tolerant, supports logging and retries, and is easily configurable to include new pilots or update intervals without downtime.

3.3 AccesS BOMS Dashboard Integration

Once the data is stored in the central database, it becomes accessible through the project's Access Suite dashboard. The dashboard provides real-time visualization and monitoring capabilities, allowing each pilot manager to explore the environmental and energy performance data of the pilot sites assigned to them by the admin in a unified and consistent interface. Thanks to the common data model enforced by the parsing framework, dashboard components can query and display measurements reliably, ensuring that data from different sources is presented in a standardized manner.

3.4 Extensibility and Future Support

The current parsing and ingestion framework is designed to be extensible. Adding a new pilot requires only the implementation of a new parser that conforms to the shared interface. All other components—including the scheduler, database schema, and dashboard—remain unchanged. This design minimizes integration overhead and supports the long-term scalability of the platform.

4 Forecasting Engine

4.1 Introduction

The Forecasting Engine is a core component of the BOMS architecture. It serves as a general-purpose forecasting system that generates predictions for all relevant measurements collected across different pilot sites. These predictions are used as inputs to the Decision Support System (DSS), enabling the generation of actionable recommendations to improve either building energy efficiency or indoor environmental conditions.

The engine automates the entire process from data ingestion to forecast delivery, including data collection, preprocessing, feature engineering, model training, validation, and forecasting. Its modular design allows support for multiple pilot sites and metrics simultaneously, providing forecasts for parameters such as temperature, humidity, CO_2 concentration, energy consumption, and solar energy production. This level of automation ensures that predictions are consistent, reproducible, and reliable across diverse building environments [1] [2].

4.2 System Architecture

The Forecasting Engine is implemented as a modular and scalable service within the BOMS framework. It is designed to support multiple pilots and metrics in parallel while maintaining a consistent and automated forecasting workflow. The architecture is composed of the following key components:

Data Interface Layer: Connects to the AccesS Suite TimescaleDB database to retrieve time-series data for all pilots and metrics. It handles data querying, aggregation, and resampling to ensure uniform input for modeling.

Preprocessing & Feature Engineering Module: Applies data cleaning, filtering, and temporal feature generation (e.g., hour of day, weekday, and seasonal cycles) to prepare datasets for model training. For energy production and consumption forecasts, this module also integrates external weather data such as temperature, solar radiation, humidity, and wind speed to capture the influence of environmental factors on building performance in terms of electrical demand and generation.

Model Management Layer: Trains and stores probabilistic forecasting models using AutoGluon's TimeSeriesPredictor with the Chronos model family. Gradient-boosting frameworks such as XGBoost are used to enhance predictive accuracy. This layer supports model versioning, retraining, and reuse for different pilots and metrics.

Evaluation & Validation Module: Assesses model accuracy using performance metrics and visual forecast comparison plots to ensure reliable predictions.

Forecast Generation & Serving Layer: Produces day-ahead and intraday forecasts, including multiple quantile predictions (p10, p50, p90), and delivers results to the Decision Support System for downstream analysis and recommendations.

4.3 Forecasting Pipeline and Workflow

4.3.1 Data Loading & Preprocessing

For each pilot and metric, the engine loads time-series data from AccesS Suite TimescaleDB database. Data is resampled to a consistent frequency (e.g., hourly), and applies metric-specific aggregation functions. Temporal features, such as sine and cosine encodings of the hour of day, weekday, and month, are added to capture seasonal patterns and trends [2]. Sensors with insufficient historical data are filtered out to ensure reliable modelling. The resulting dataset is then structured into a standardized time-series format ready for model training.

Data quality checks are applied to handle missing values, outliers, and sensor anomalies through techniques like forward/backward filling and z-score filtering. Sensors with insufficient historical coverage or inconsistent sampling rates are excluded to ensure robust model training.

4.3.2 Model Training

Probabilistic models are trained independently for each pilot, metric, and monitored space using AutoGluon's Chronos model family [3]. Each model is designed to capture the specific dynamics of the corresponding metric, whether it is temperature, CO_2 concentration, energy consumption, or solar production, and generates forecasts up to three days ahead at an hourly resolution.

The training process leverages temporal covariates such as hour of day, weekday, and month, as well as external features like weather data, incorporated through gradient-boosting frameworks such as XGBoost [4]. These covariates allow the model to capture complex nonlinear relationships and interactions that influence the target metric.

Hyperparameter optimization is performed automatically within AutoGluon to select the best combination of model parameters for each metric and pilot site. This includes tuning aspects such as learning rate, tree depth, number of estimators, and regularization coefficients in gradient-boosted models. AutoGluon's automated framework also supports ensemble learning, where multiple base models are combined to improve predictive performance and robustness.

The pipeline is fully parallelized, allowing simultaneous training of multiple models across different pilots, metrics, and monitored spaces. Each trained model is stored individually per pilot and metric, enabling incremental updates as new data becomes available, while also supporting model versioning to maintain a record of the training history.

4.3.3 Testing & Validation

Forecast models are evaluated using a held-out test set, with predictions compared against actual measurements. For building-level energy forecasts, the testing set covers the last three days of available data. Key performance metrics include Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Symmetric Mean Absolute Percentage Error (SMAPE), and Weighted Absolute Percentage Error (WAPE), providing a comprehensive view of forecast accuracy in both absolute and relative terms [5].

Mean Absolute Error (MAE) measures the average magnitude of absolute errors in the same units as the target metric:

$$MAE = (1/n) * \Sigma |y\_i - \hat{y}\_i|$$

Advantages: Simple to interpret; directly reflects absolute deviations. *Limitations:* Sensitive to scale; large values dominate the metric.

Mean Absolute Percentage Error (MAPE) measures the average absolute percentage error:

$$MAPE = (100/n) * \Sigma |y\_i - \hat{y}\_i| / y\_i$$

Advantages: Expressed as a percentage, allowing comparison across metrics of different scales. *Limitations:* Sensitive to small or zero actual values, which can inflate error percentages.

Symmetric Mean Absolute Percentage Error (SMAPE) normalizes the error to treat over- and under-forecast deviations symmetrically:

$$SMAPE = (100/n) * \Sigma |y\_i - \hat{y}\_i| / ((|y\_i| + |\hat{y}\_i|)/2)$$

Advantages: Reduces bias when forecasts consistently over- or under-estimate. *Limitations:* Can be unstable for very low actual and forecasted values.

Weighted Absolute Percentage Error (WAPE) is a custom metric used for solar energy production, which weights forecast errors according to actual values, giving more importance to periods of high production:

$$WAPE = 100 * \Sigma (|y\_i - \hat{y}\_i| / y\_i * y\_i / \Sigma y\_j)$$

Advantages: Emphasizes errors during high-production periods, avoiding distortion from low-generation periods. *Limitations:* Less intuitive than standard metrics; specific to applications where weighting by actual values is meaningful.

Forecast plots are also generated to visually inspect performance and detect potential deviations or trends. These plots actual and predicted values over the forecast horizon. The following plots present two forecasting examples: one at the building level and another at the room level.

The first plot compares actual and forecasted values for the total load demand of a pilot over the last three days. The second plot presents a similar comparison at the room level, focusing on CO2 levels across different room areas within the same building.

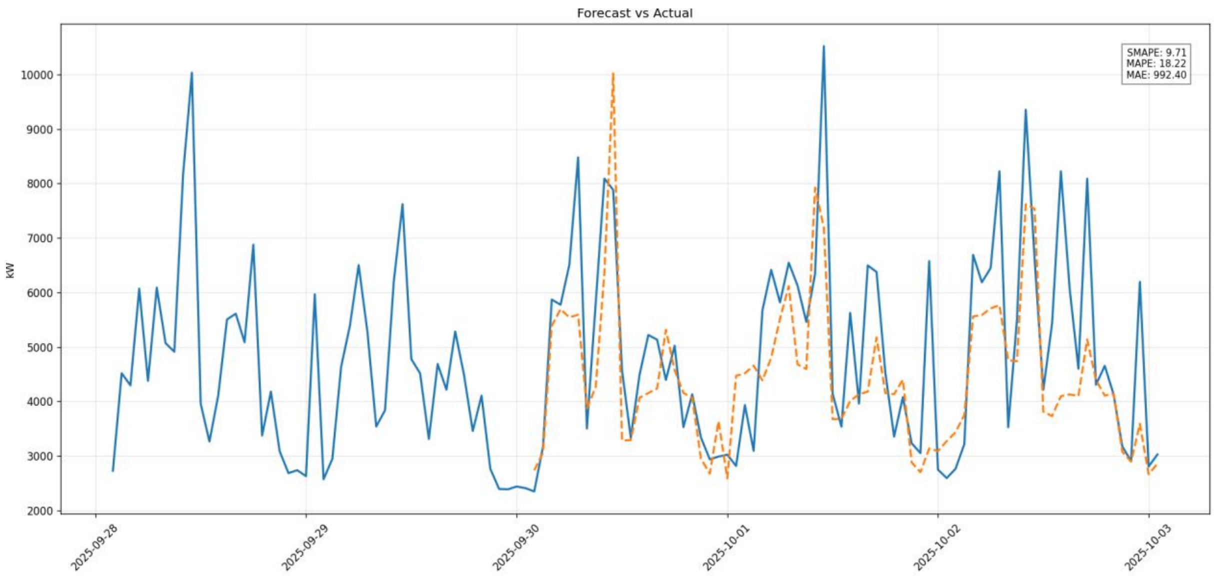

Figure 3. Building-level load demand forecast showing actual measurements (solid line) versus model predictions (dashed line).

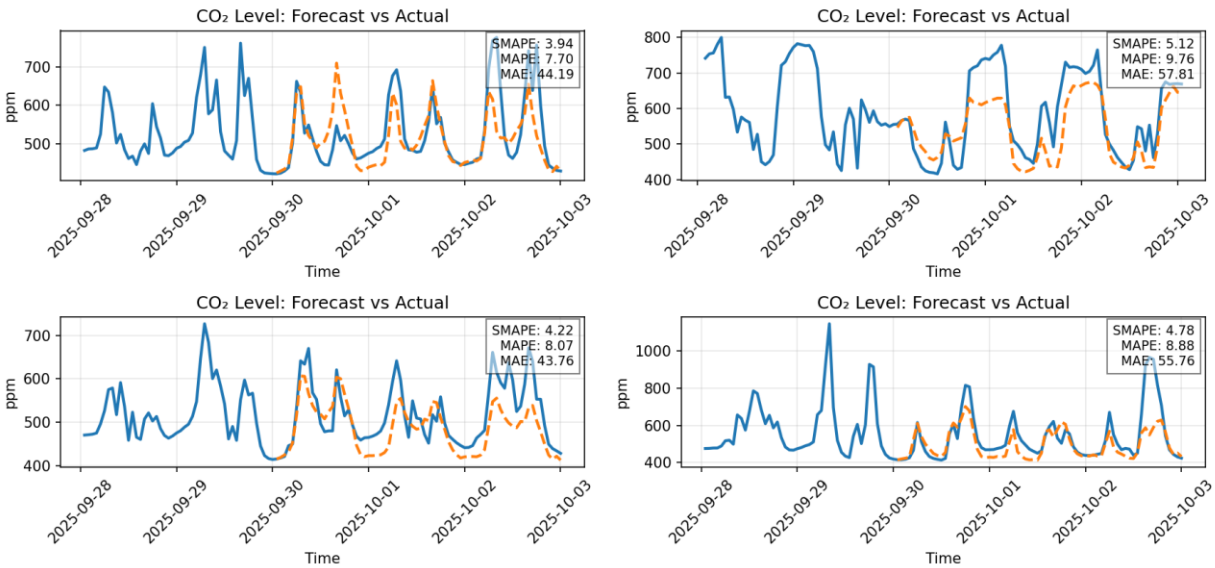

Figure 4. Room-level C02 Level forecast showing actual measurements (solid line) versus model predictions (dashed line).

4.3.4 Real-Time Forecast Generation

Once trained and validated, models can produce forecasts in different modes: **day-ahead** (next 24 hours) or **intraday** (from the current time until midnight). Forecasts include multiple quantiles (p10, p50, p90) to show the range of possible outcomes and support informed decision-making. This functionality allows the system to provide actionable, room-level forecasts of indoor conditions, as well as building-level energy and solar production predictions, supporting optimized management of comfort, energy efficiency, and sustainability. A scheduler is deployed to automatically trigger for forecasting engine at predefined intervals, day ahead forecast is generated at midnight, while intraday forecasts run on an hourly basis.

The forecasts are displayed in real time on the BOMS dashboard within a dedicated "Sensor Forecasts" tab, allowing users to monitor predictions at both the building level and across multiple rooms.

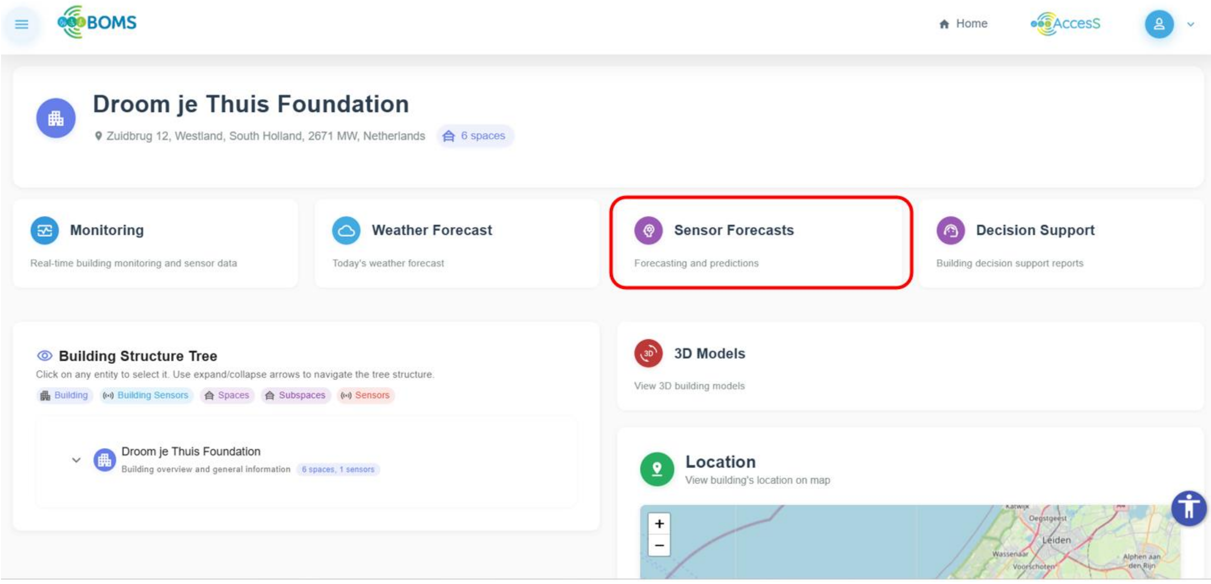

Figure 5. Sensor Forecasts tab for a Droom je Thuis pilot

After entering the tab, the user can view real-time forecast results for any selected sensor and metric, with the option to display predictions for a specific day or the current time.

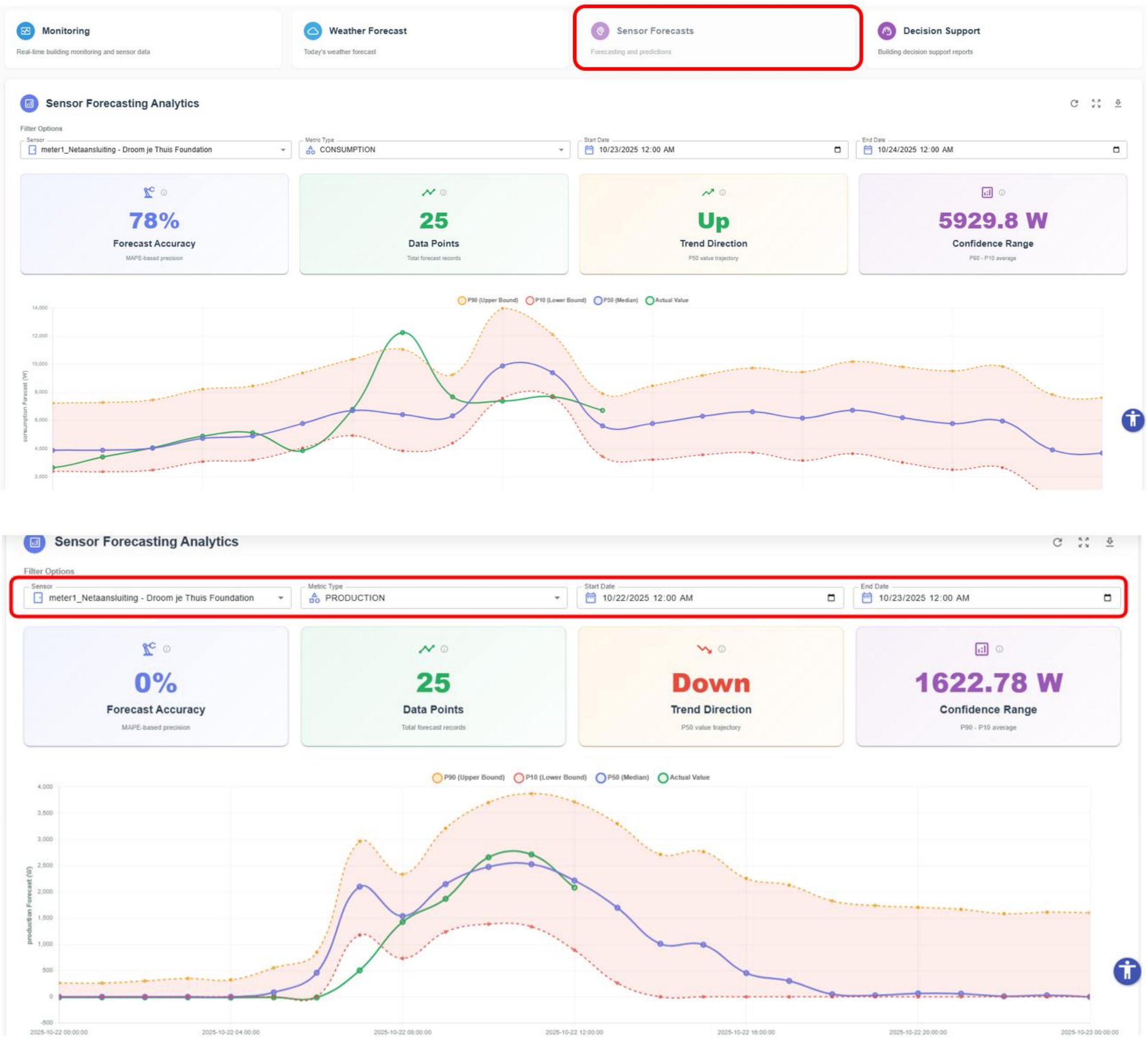

Figure 6. Real-time forecast view, showing predicted and actual values for a selected sensor and metric, including confidence intervals (p10-p90).

4.4 Technical Stack

The Forecasting Engine incorporates several advanced features to ensure flexibility, scalability, and precision across multiple pilot sites and metrics. Each pilot site and metric are processed independently but simultaneously, enabling efficient parallelization and allowing the system to handle multiple buildings and sensors concurrently. The models can generate forecasts of variable lengths, ranging from hourly predictions up to 72 hours ahead, configurable per metric or per pilot depending on operational requirements.

Probabilistic forecasts include p10, p50, and p90 quantiles, providing a range of possible outcomes to support risk-aware decision-making. Forecast outputs are structured in JSON format, including timestamped quantiles, which enables integration with DSS agents for automated action recommendations. In addition, the engine implements continuous learning, with incremental retraining scheduled as new data becomes available. This ensures models maintain accuracy and adapt to evolving building behavior or sensor characteristics.

The Forecasting Engine is implemented in Python, leveraging open-source libraries for data processing, modeling, and evaluation. The main components and frameworks include:

Data Handling: Time-series manipulation and database interaction with the AccesS Suite TimescaleDB are performed using pandas[3] and SQLAlchemy[4].

Feature Engineering & Pre-processing: Feature extraction, normalization, and outlier detection are carried out with scikit-learn[5] and numpy[6].

Forecasting Framework: Probabilistic forecasting is implemented using AutoGluon's TimeSeriesPredictor[7] in combination with the Chronos model family[8].

Gradient-Based Models: External covariates are incorporated and model accuracy is boosted with XGBoost[9].

Visualization & Evaluation: Forecasts and performance metrics are visualized using matplotlib[10].

Automation & Scalability: Scheduling and automation of forecast generation and model retraining are managed by APScheduler11, enabling fully automated and time-based execution of forecasting tasks. A separate scheduler is created and assigned for each pilot to ensure independent and efficient processing. The engine is containerized and deployed using Docker12, ensuring consistent operation and easy integration within the BOMS infrastructure.

[3] pandas: https://pandas.pydata.org/

[4] SQLAlchemy: https://www.sqlalchemy.org/

[5] scikit-learn: https://scikit-learn.org/

[6] NumPy: https://numpy.org/

[7] AutoGluon: https://auto.gluon.ai/

[8] Chronos: https://github.com/awslabs/chronos-forecasting

[9] XGBoost: https://xgboost.readthedocs.io/

[10] matplotlib: https://matplotlib.org/

[11] APScheduler: https://apscheduler.readthedocs.io/

[12] Docker: https://www.docker.com/

5 Decision Support System

5.1 Introduction

Buildings are complex ecosystems that significantly influence human well-being, energy consumption, and environmental sustainability. As global efforts intensify toward reducing carbon emissions and optimizing energy use, maintaining high indoor air quality (IAQ) while achieving energy efficiency has become a critical challenge. Traditional building management systems rely on static control strategies and manual interventions that often fail to adapt to dynamic environmental conditions or occupant needs. Recent advances in agentic artificial intelligence (AI)—AI systems capable of autonomous decision-making, collaboration, and continuous learning—are transforming how buildings can sense, interpret, and act upon environmental data [6] [7] [8] [9]. By integrating multi-agent frameworks with real-time sensing, predictive modeling, and adaptive control, agentic AI enables buildings to function as self-optimizing entities: improving air quality, reducing energy waste, and enhancing occupant comfort. This convergence of intelligent agents and sustainable building operation marks a paradigm shift toward autonomous, human-centric, and energy-efficient smart buildings.

The Decision Support System (DSS) is a core subcomponent of the Building Operations Management System (BOMS), designed to provide intelligent building monitoring and management. As outlined in **D1.2**, which describes the system architecture, the DSS plays a central role in enabling data-driven decision-making within BOMS. The DSS leverages Agentic AI to orchestrate specialized agents that analyse building conditions, detect inefficiencies, and identify anomalies in real time. Its main functions include:

- **Indoor Environmental Analysis** – detecting comfort issues and deviations from optimal conditions.
- **Energy Efficiency Analysis** – evaluating consumption patterns and recommending cost-optimized strategies.
- **Anomaly Detection** – identifying faulty or drifting sensors via configurable thresholds.

By combining real-time sensor data, forecasting models, and building standards, the DSS generates actionable insights and recommendations, assisting facility managers and building operators in ensuring comfort, energy efficiency, and operational reliability.

5.2 System Architecture

The architecture of DSS is organized into several layers that cooperate to provide end-to-end decision support. The data layer collects and stores real-time measurements from sensors deployed throughout the building. These include environmental data such as temperature, humidity, and air quality, as well as energy-related data such as consumption and generation. Historical records are maintained to allow trend analysis and the training of predictive models. On top of this, the forecasting engine generates short-term projections for each monitored variable, enabling the system to anticipate how conditions will evolve rather than relying solely on instantaneous measurements.

The agentic AI layer implements the reasoning process of the system. It is built with LangGraph, which manages the orchestration of multiple domain-specific agents [9], and the open-source Qwen3 model, which supports the analysis and natural language generation required for producing reports. The outputs of the agents are collected by the decision support layer, which aggregates their findings and synthesizes a coherent narrative about the state of the building. This information is then presented through a user interface that allows operators to review current conditions, explore forecast-based scenarios, and consult detailed reports.

5.3 Prompt Engineering Analysis

The Decision Support System employs multiple autonomous agents—each responsible for specific analytical domains such as **energy optimization** and **indoor conditions and comfort analysis**—that operate through structured prompt engineering. In this context, prompt engineering is not limited to formulating queries but is a **core mechanism for defining agent behavior**, data interpretation logic, and the structure of the generated outputs. Each agent is designed with prompt templates that specify input parameters, reasoning steps, and output schemas (e.g., JSON), ensuring deterministic and interpretable responses aligned with the system's analytical objectives.

For the **Energy Optimization Agent**, prompts are designed to process time-series energy data, dynamic pricing information, and building consumption profiles. The prompt explicitly requests the agent to identify **peak and off-peak periods**, calculate **total cost and potential savings**, and propose **actionable recommendations** such as load shifting or demand response strategies. The structured JSON output reflects the precision of the prompt—containing fields like `peak_hours`, `average_price`, `savings_percentage`, and `recommendations`, each populated through controlled natural language reasoning. This structure allows seamless integration with downstream processes such as dashboard visualization, automation triggers, or cost-benefit analytics.

Similarly, the **Indoor Conditions and Comfort Agent** relies on prompt templates that define the evaluation logic for environmental sensor data, including CO_2, PM2.5, and thermal parameters. The prompt guides the agent to classify room conditions (`state`) based on threshold rules, detect anomalies, and generate targeted recommendations for corrective actions. For example, when CO_2 levels exceed acceptable limits, the agent responds with a "Poor" state and ventilation improvement suggestions, demonstrating context-aware reasoning driven by prompt structure.

Technically, prompt engineering in this system enforces **consistency, domain alignment, and semantic control** across all agents. By embedding rule-based reasoning and output formatting instructions directly within the prompts, the system ensures reproducible analytical behavior and easy integration into existing decision workflows. The result is a scalable and adaptive multi-agent architecture where prompt design governs both **analytical depth** and **operational reliability**, enabling human-understandable insights to be produced from complex environmental and energy datasets in real time.

5.4 Data Sources

5.4.1 Environmental Data

The DSS uses environmental data from a network of sensor measurements installed in the pilot sites, with sensors deployed in each individual space (room). These sensors monitor key indoor comfort and air quality parameters, including CO_2, VOC, $PM_{2.5}$, PM_{10}, PM_2, temperature, humidity, and NOx levels. This data provides a high-resolution understanding of indoor environmental conditions and occupant comfort, forming the basis for real-time monitoring, anomaly detection, and optimization of indoor comfort levels.

5.4.2 Weather Data

The DSS integrates weather forecast data obtained from the OpenWeather API. This data source supplies short- and medium-term forecasts of outdoor conditions such as temperature, humidity, wind speed, and solar radiation. Incorporating this information allows the DSS to anticipate changes in external conditions and adjust building operation strategies accordingly, supporting predictive control and energy efficiency.

5.4.3 Energy Price Data

To incorporate economic considerations into its decision-making process, the DSS retrieves day-ahead market energy price data from the ENTSO-E API[13]. This enables the system to analyze market dynamics, evaluate cost implications, and recommend cost-optimized energy management strategies, ensuring that operational decisions take into account both energy efficiency and price variability.

5.4.4 Energy Consumption and Generation Data

The DSS also processes energy consumption and generation measurements collected from energy meters installed in the pilot sites. These datasets provide real-time and historical insights into building energy performance. In combination with forecasted consumption and generation data, they enable the DSS to assess system efficiency, identify operational inefficiencies, and propose optimized control actions that balance comfort, sustainability, and cost-effectiveness.

5.5 Core Components

5.5.1 Indoor Conditions and Comfort Agent

The first domain is dedicated to indoor environmental quality and occupant comfort. The Comfort Agent evaluates the most recent sensor measurements together with forecasted values for temperature, humidity, CO_2 levels, and other relevant parameters. These are compared against global standards such as WHO and ISO, ensuring that the assessment aligns with established benchmarks of comfort. Each space in the building is assigned a condition category, which may be excellent, good, acceptable, or poor. The system does not only assess the current state but also considers the forecasts to identify potential threats such as an expected rise in indoor temperatures or deteriorating air quality. The agent produces a descriptive report for each space that explains the classification, describes the issues identified, and recommends actions to improve or maintain comfort [9].

The indoor conditions analysis process within the Decision Support System (DSS) begins with a user configuration step, where the user selects the environmental metrics to be included in the analysis and report — such as CO_2, VOC, $PM_{2.5}$, PM_{10}, temperature, humidity, and NOx. The user can also choose whether to include forecasted data in the analysis. When this option is enabled, the DSS leverages predictive modeling to operate in a predictive maintenance mode, identifying potential future comfort deviations or indoor air quality issues before they occur. This proactive approach supports facility managers in maintaining optimal indoor conditions, improving occupant comfort, and preventing degradation of environmental quality over time.

The following image illustrates the configuration window of the Indoor Conditions module, where the user can select the environmental metrics to be included in the indoor conditions analysis.

[13] https://transparency.entsoe.eu/dashboard/show

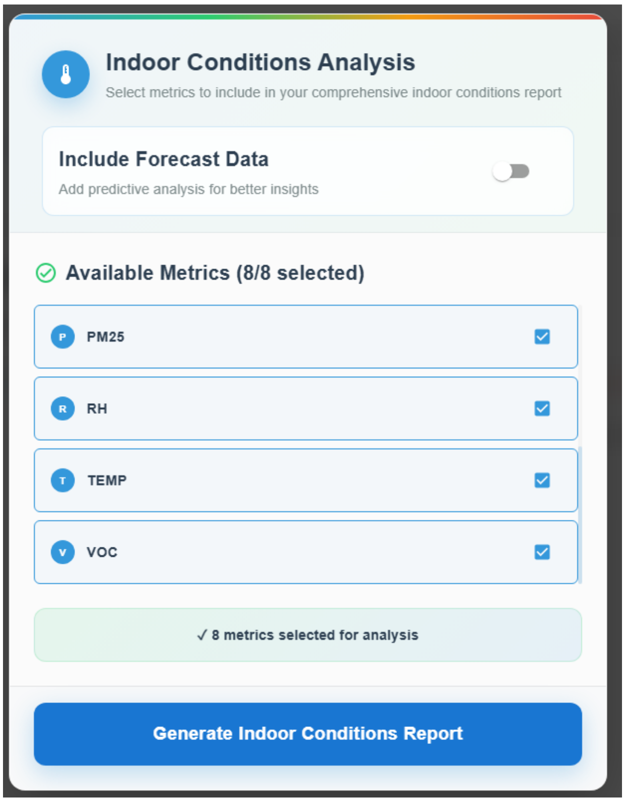

Figure 7. Indoor Conditions Configuration

Table 10 provides a clear description of the workflow for the indoor conditions analysis engine, from start to the generation of the final report:

Table 10. Indoor Conditions Assessment Workflow

Step	Description	Inputs	Outputs
1.User Request	A facility manager or operator triggers the assessment for a specific room through the DSS interface. The user may choose whether to include forecasted data.	- User command (room ID, time range) - Optional flag: include forecast	Execution request
2.Data Retrieval	The agent fetches the latest indoor condition measurements from the database. If the user requested forecasts, it also retrieves the corresponding forecast reports.	- Real-time sensor data - (Optional) Forecasted measurements	Raw data (actual values, plus forecasts if selected)
3.Data Pre-processing	Synchronizes actual and optional forecasted values, resolves missing entries, and prepares features for AI-based evaluation.	- Raw sensor and forecast data (if included)	Clean, structured dataset

4.Comfort Analysis	Evaluates the dataset against established comfort standards (e.g., WHO[14]) using AI reasoning to detect issues.	- Preprocessed dataset - Comfort thresholds	Comfort status per parameter (Excellent/Good/Acceptable/Poor)
5. Assessment Generation	Produces a structured, human-readable assessment for the requested room, summarizing comfort conditions and deviations.	- Comfort analysis results	Final per-room assessment report

The final report produced by Indoor Conditions Component provides a comprehensive summary of the indoor environmental conditions across all monitored spaces. At a high level, it presents the number of rooms identified as having potential issues versus those in a good state, offering a quick overview of the building's overall comfort and air quality performance. The report also includes a detailed breakdown for each individual room, describing the current environmental state, highlighting deviations from optimal conditions, and providing recommendations to preserve or improve the indoor environment. These recommendations support facility managers in taking targeted actions to maintain occupant comfort, ensure air quality, and optimize energy efficiency throughout the building.

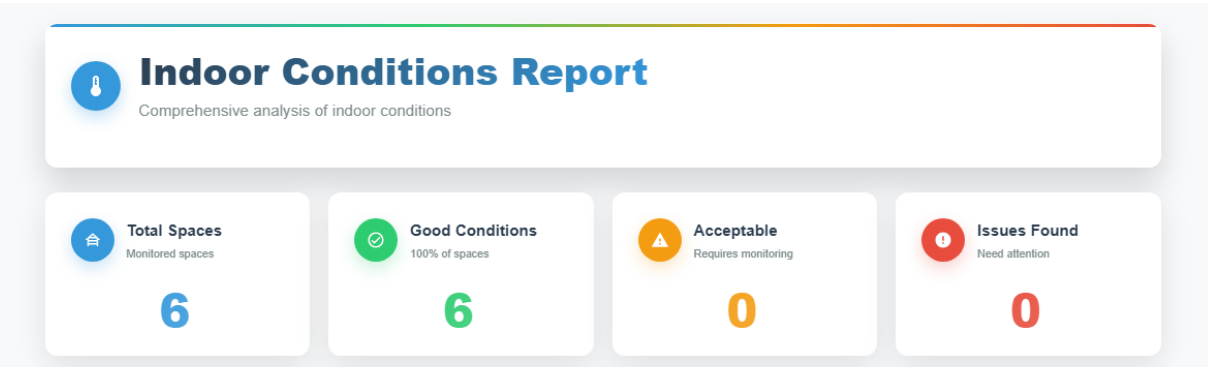

Figure 8. General Report of each Space

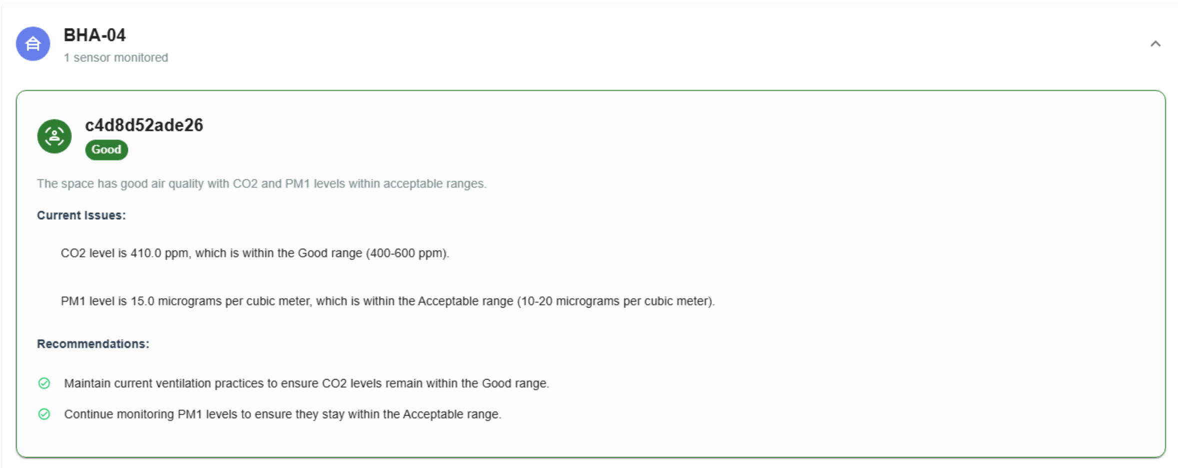

Figure 9. Example of Space analysis

[14] https://www.who.int/

5.5.1.1 Indoor Conditions Report Scheme

The Indoor Conditions Report Schema defines the output data structure produced by the Indoor Environment Assessment Agent. This agent monitors, diagnoses, and forecasts the quality of indoor spaces based on sensor data to generate actionable recommendations for maintaining optimal conditions and occupant comfort. The report provides a comprehensive overview of each monitored space, including its current state, detected issues, potential future problems, and suggested interventions. It is designed for seamless integration with smart building management systems, facility dashboards, and decision-support tools.

Table 11. Indoor Conditions Report Scheme

Field	Type	Description	Example
spaceName	str	The name or label of the space within the building model.	"Meeting Room A-203"
sensorID	str	Unique identifier of the sensor or sensor group monitoring this space.	"sensor_4h29b8d"
state	str	The current condition or operational state of the space, derived from sensor readings.	"Suboptimal"
issues	List[str]	List of current detected issues affecting indoor quality (e.g., temperature, CO_2, humidity, VOC levels).	["High CO_2 concentration", "Low air circulation"]
description	str	A human-readable summary of the current space conditions and detected anomalies.	"CO_2 levels have exceeded 1200 ppm, and humidity is below comfort range."
future_issues	List[str]	Forecasted problems predicted by the system based on trend analysis or ML models.	["Possible temperature rise due to HVAC load imbalance", "Increased CO_2 expected during occupancy peak at 14:00"]
recommendations	List[str]	List of recommended actions or system adjustments to maintain or restore optimal indoor conditions.	["Increase ventilation rate by 10%", "Schedule HVAC inspection", "Adjust humidity control setpoint"]

5.5.2 Energy Efficiency Agent

The second domain addresses the energy performance of the building. The Energy Efficiency Agent analyzes the balance between energy consumption and generation, when available, in relation to day-ahead energy prices. Forecasted values of consumption and generation are used to anticipate the future load profile of the building and to evaluate how it aligns with the expected price signals. The purpose of this analysis is to detect periods of high energy cost and to recommend strategies for optimizing building operations. This may include proposing occupancy schedules that align with low-cost periods or advising load shifting to reduce expenses. By combining real consumption and generation forecasts with day-ahead energy prices, the agent helps operators make informed decisions that minimize costs and improve sustainability [6] [10]. The result is a report that highlights inefficiencies, describes opportunities for savings, and provides recommendations for cost-effective energy management.

The following figures illustrate the user interface of the decision support system developed for energy optimization in buildings. The interface provides an integrated overview of energy consumption patterns, costs, and potential savings, distinguishing between peak and off-peak periods. It visualizes key performance indicators such as total consumption, average energy price, and estimated savings potential, enabling quick assessment of current energy performance. Furthermore, the system offers actionable recommendations—such as optimal scheduling of high-energy activities and load shifting strategies—to enhance efficiency and reduce operational costs. Together, these components demonstrate the system's capability to support informed decision-making through data-driven energy management insights.

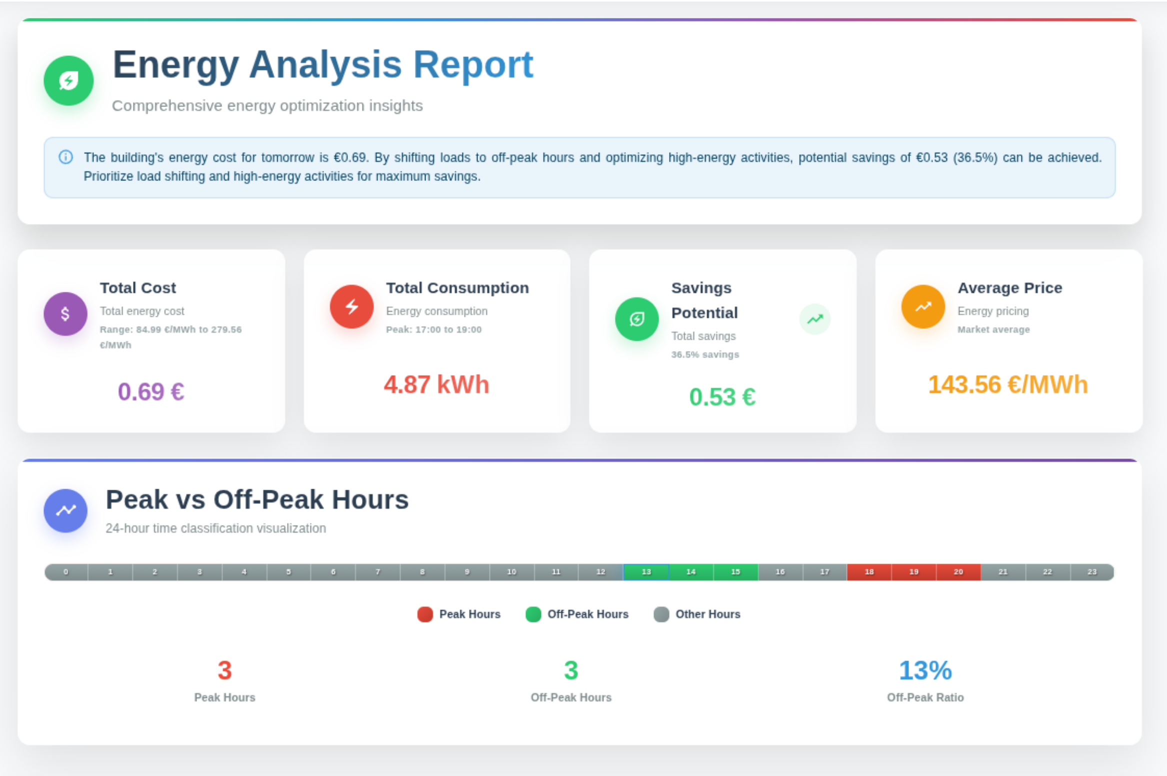

Figure 10. Energy Analysis Report

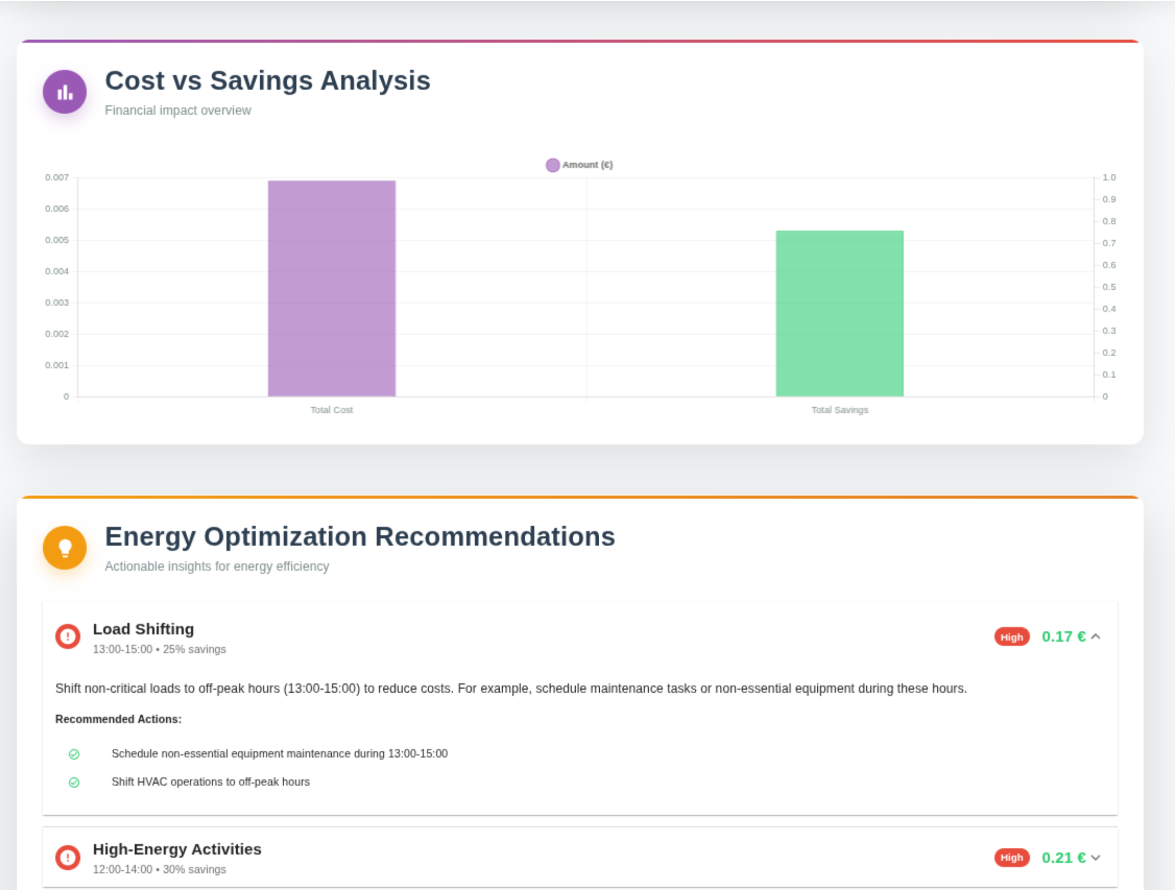

Figure 11. Energy Optimisation Recommendations

The following workflow describes etc… or something like that

Table 12. Energy Optimisation Workflow

Step	Description	Inputs	Outputs
1. User Request	A facility manager or operator triggers the energy optimization assessment through the DSS interface. The user may choose whether to include forecasted consumption.	- User command (building ID, time range - Optional flag: include forecast	Execution request
2. Data Retrieval	The agent collects day-ahead market price data and retrieves historical consumption. If selected, forecasted consumption data is also included.	- Day-ahead electricity price data - Historical energy consumption (Optional) Forecasted consumption	Raw dataset (price + load profiles)
3.Data Preprocessing	Aligns price and consumption data on the same temporal scale, handles missing entries,	- Raw datasets (prices, historical, forecast)	Clean, structured dataset

	and normalizes profiles for analysis.		
4.Efficiency Analysis	Evaluates the relationship between consumption patterns and price signals, identifies cost-saving opportunities (e.g., load shifting, peak shaving), and detects inefficiencies in operations.	- Preprocessed dataset - Energy efficiency models	Efficiency indicators (KPIs, deviations, potential savings)
5.Recommendation Engine	Generates actionable insights, such as rescheduling loads, optimizing HVAC operation, or leveraging storage to reduce cost and improve efficiency.	- Efficiency analysis results	Energy optimization recommendations
6.Report Generation	Produces a structured report summarizing current efficiency, expected costs, and recommended operational strategies.	-Recommendations -Efficiency indicators	Building-level energy efficiency report

5.5.2.1 Energy Optimization Report Scheme

The Energy Optimization Report Schema defines the output data structure produced by the Energy Optimization Agent. This agent analyzes energy consumption, pricing data, and usage patterns to generate actionable recommendations for improving efficiency and reducing costs. The report provides a complete overview of energy usage behavior, pricing dynamics, and optimized strategies for load management and cost reduction. It is intended for seamless integration with smart building management systems, facility dashboards, and decision-support tools. The table below outlines the data included in the Energy Optimization Report:

Table 13. Energy Report Scheme

Field	Type	Description	Example
Price Overview			
peak_hours	str	Time period of high energy prices.	"10:00–14:00"
off_peak_hours	str	Time period of reduced energy prices.	"22:00–06:00"
average_price	float	Average electricity price (€/MWh).	78.2
price_range	str	Range of observed prices during the analysis period.	"€45–120/MWh"
Consumption Overview			

Field	Type	Description	Example
total_consumption	float	Total building energy consumption during the analysis window (kWh).	5243.6
total_cost	float	Total energy cost (in euros).	678.25
peak_consumption_time	str	Time of maximum energy use.	"14:00"
low_consumption_time	str	Time of minimum energy use.	"03:00"
Recommendations			
recommendations	List [EnergyRecommendation]	A list of actionable energy-saving suggestions.	[EnergyRecommendation(...)]
Summary			
total_savings	float	Projected total cost savings (in euros).	112.45
savings_percentage	float	Total percentage reduction achievable in energy costs.	16.8
top_actions	List[str]	The three most impactful recommended actions.	["Shift HVAC load to off-peak", "Optimize lighting schedules", "Reduce standby consumption"]
Metadata			
building_id	str	Identifier for the building under analysis.	"Bruco Huisvesting Adviseurs B.V."
analysis_date	str	ISO 8601 date when the analysis was performed.	"2025-10-24"
summary	str	A concise narrative summarizing the key findings and recommendations.	"Energy costs can be reduced by 16.8% by shifting high-load activities to off-peak hours and optimizing HVAC operation."

The table below details the data contained in the Recommendation Report, which forms part of the Energy Optimization Report:

Table 14. Recommendation Scheme

Field	Type	Description	Example
title	str	The title or summary of the recommendation.	"Peak Hour Load Reduction"

Field	Type	Description	Example
description	str	A detailed explanation of the action to take.	"Shift non-essential loads to off-peak hours to reduce peak-time costs."
time_window	str	The relevant time period for implementing the recommendation.	"10:00–14:00"
priority	str	The urgency or importance of the action (High, Medium, Low).	"High"
savings_percentage	float	The estimated energy savings as a percentage of total consumption.	15.5
savings_cost	float	The projected cost savings in euros.	45.30

5.5.3 Anomaly Detection Tool

The third domain focuses on the integrity and reliability of the monitoring system itself. The Anomaly Detection Agent inspects all incoming sensor data and applies manually defined thresholds for each sensor type. When readings deviate from expected ranges, the system flags them as anomalies. Such anomalies may indicate equipment malfunction, calibration drift, or unexpected conditions within the monitored environment. The agent generates alerts with descriptions of the detected issues and suggests corrective actions, such as sensor maintenance or recalibration, ensuring that decision-making is based on reliable data [8].

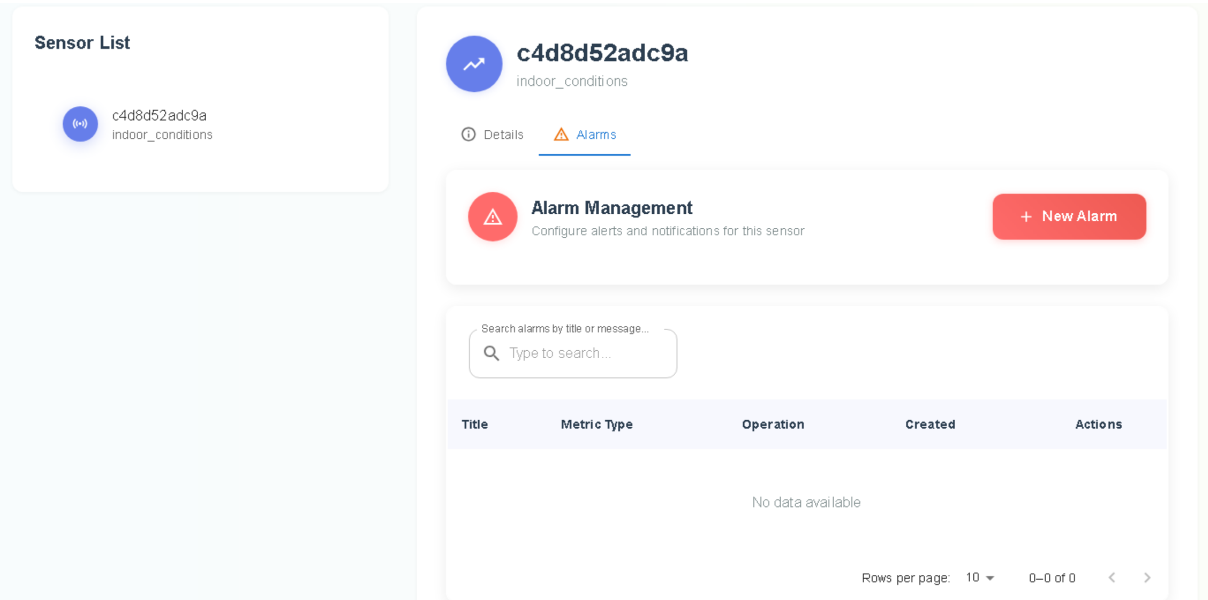

Figure 12 Manual Alarm Generation

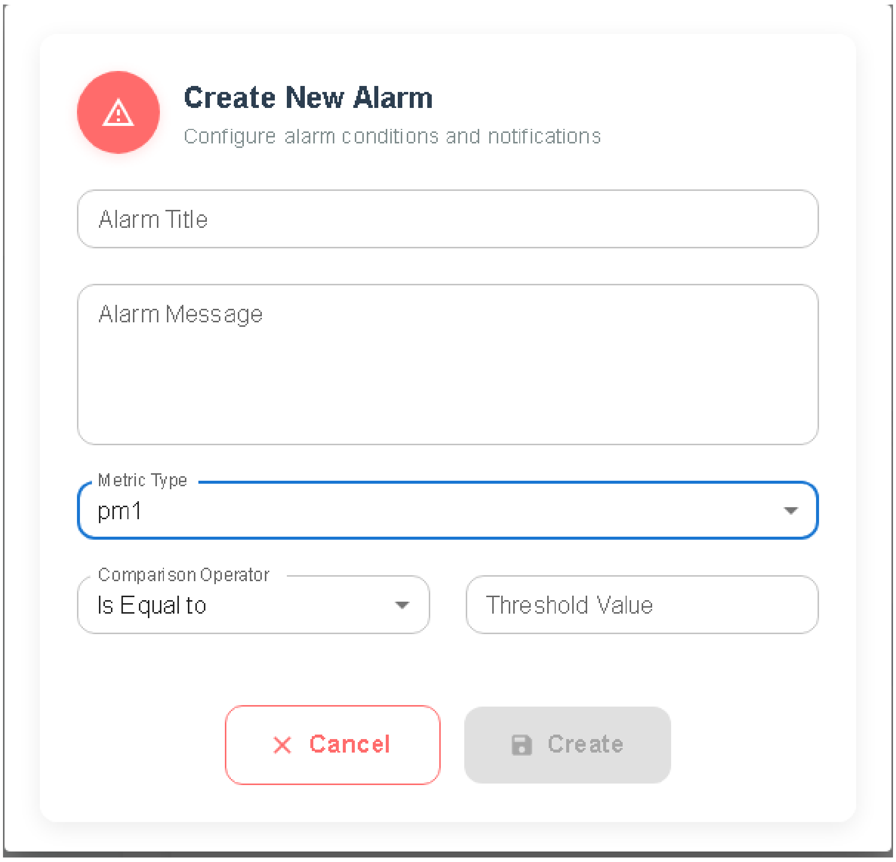

Figure 13 Alarm Configuration

5.6 Decision Support Workflow

The DSS workflow begins with comprehensive data collection, retrieving the latest sensor readings alongside forecast data, while also incorporating external signals such as weather predictions and day-ahead energy prices. The Comfort Agent, Energy Efficiency Agent, and Anomaly Detection Agent independently analyze this data, and their results are aggregated and processed through LangGraph, with the Qwen3 model generating a structured, human-readable narrative. This multi-agent approach ensures that comfort, energy efficiency, and anomaly detection are addressed in a coordinated and explainable manner.

By integrating forecasts into its assessments, DSS enables a proactive approach to building monitoring and management, allowing operators to anticipate issues before they occur rather than reacting after the fact. The system produces detailed reports for each space and the building as a whole, outlining the current state, detected issues, and actionable recommendations. Reports can be generated on demand or scheduled regularly, providing continuous oversight of building operations and guiding operators toward better decisions that enhance occupant comfort, reduce costs, and maintain the reliability of sensor systems.

5.7 Technical Stack

DSS is implemented as a modular system that integrates advanced AI techniques with robust data management. The agentic framework is built on LangGraph[15], which allows the orchestration of

[15] https://www.langchain.com/langgraph

specialized agents. Reasoning and natural language generation capabilities rely on the Qwen3-14B[16] model, running on an RTX 4080 GPU and supporting a maximum length of 40k tokens. This model was selected because it provides efficient inference on available hardware and fully open-source licensing, which supports transparency, reproducibility, and compliance with Horizon Europe requirements. Data ingestion and forecasting pipelines are implemented in Python and can accommodate both statistical and machine learning-based models. Comfort evaluations rely on internationally recognized standards, including WHO[17] guidelines. Deployment can be achieved using containerized microservices, allowing for scalability and flexibility. The system exposes a REST API and provides a web-based dashboard for interaction with building operators.

[16] https://qwen.ai/research

[17] https://www.who.int/

6 Analytics & Visualization

6.1 Overview

The BOMS dashboard offers a user-friendly and interactive platform for real-time monitoring and analysis of building operations. Through the integration of smart sensors, IoT devices, and other digital tools within the AccesS Suite, the system collects and visualizes data related to indoor environmental conditions, energy consumption, and renewable energy production. The dashboard transforms complex data into clear, actionable insights, enabling building operators to make informed decisions that enhance comfort, energy efficiency, and sustainability.

To ensure that the insights provided by the BOMS dashboard are securely accessed and tailored to each user's responsivities, the system incorporates a robust authentication and authorization mechanism. This security layer guarantees that users only see data relevant to their assigned buildings and roles, maintaining confidentiality and compliance with organization policies.

6.2 Authentication System Integration

BOMS is securely integrated with Keycloak[18], an enterprise-grade identity and access management platform. This integration uses the Keycloak instance hosted at **auth-access-suite.iti.gr**, within the **access-suite-realm**, where **boms-client** is registered as the dedicated client application. Keycloak fully handles both authentication and authorization, providing:

- Token-based security – ensuring secure, efficient communication between BOMS and its backend services.
- Single Sign-On (SSO) – allowing users to log in once and gain access to all integrated services.
- Centralized identity governance – enabling consistent security and simplified user management.

BOMS dynamically leverages the pilot information embedded in each authenticated user's access token to determine their scope of data access. This information is processed in collaboration with the backend API, which filters and serves only the operational data relevant to the facilities assigned to the user. This design guarantees strict data isolation and role-based access per building, aligning with organization security policies. Access control is further refined through clearly defined **user roles**, each with specific permissions:

- **Admin** – Full access across all pilots and buildings.
- **Building Manager** – Access limited to the pilots and buildings assigned to them.
- **Engineer** – Technical role with access limited to their assigned pilots and facilities.
- **User** – View-only access, restricted to their assigned pilots and facilities.

This integration provides a robust, scalable, and policy-compliant authentication and authorization framework, ensuring that only authorized users can access relevant operational data within BOMS.

6.3 Interactive Data Insights

6.3.1 Monitoring Page

The Building Overview page illustrated in Figure 14 serves as the main entry point to the BOMS Dashboard, providing users with an overview of all monitored buildings within the system. Each

[18] https://www.keycloak.org/

building card represents a pilot site and allows users to access and monitor real-time data for buildings where they have been granted viewing rights by the administrator.

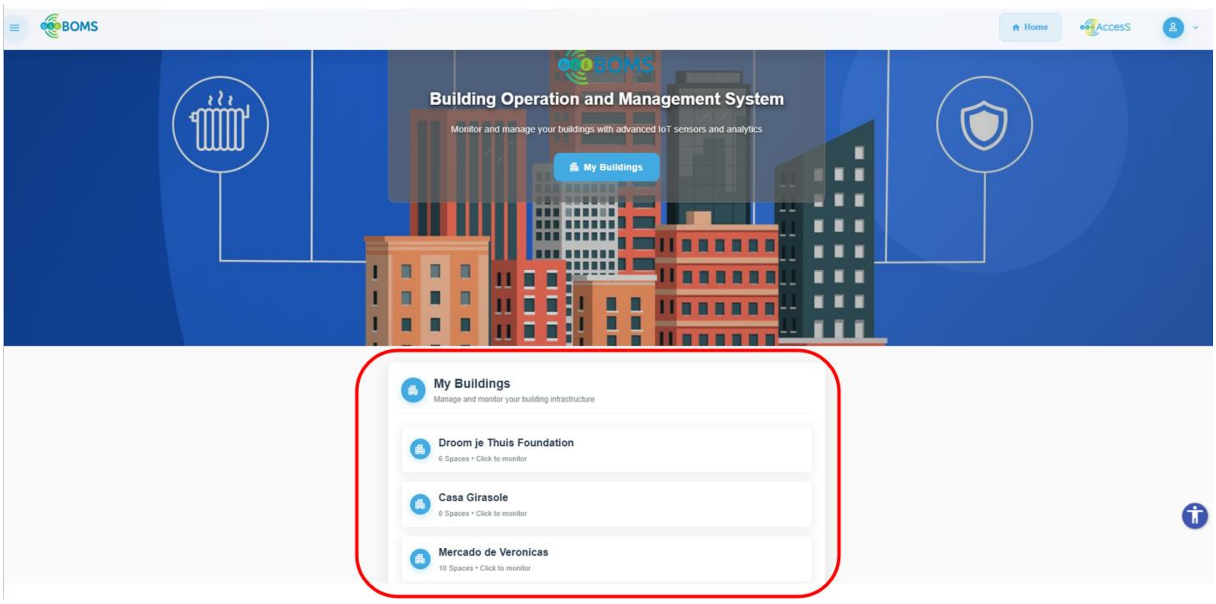

Figure 14 AccesS Building Overview Page

Once the user chooses to enter a respective building as in Figure 15, the Building Overview page provides access to key modules of the BOMS platform. Users can navigate to Monitoring for real-time sensor data, Weather Forecast for current weather conditions, Sensor Forecasts for predictive analytics, and Decision Support for generating building performance and management reports. This interface acts as the central hub for exploring and managing detailed building information.

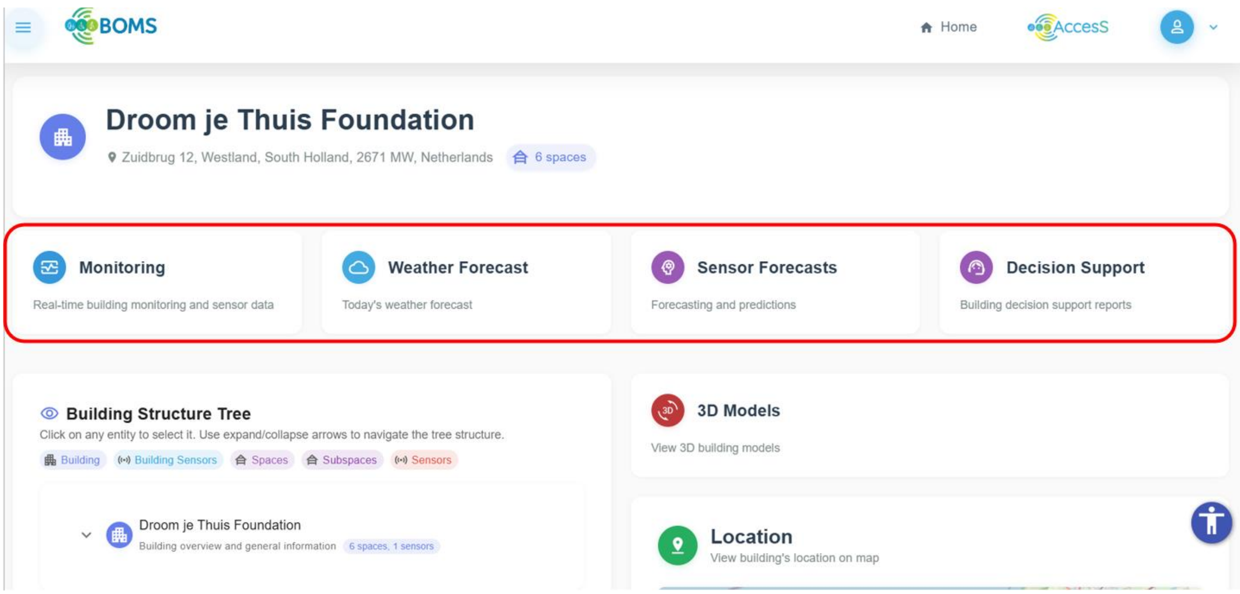

Figure 15 BOMS Dashboard

The dashboard provides real-time monitoring of key building parameters such as CO_2 levels, temperature, energy use, solar output, and air quality indicators (PM1, PM2.5, PM10, and humidity). Interactive charts let users explore data across different spaces and time periods, making it easy to understand building performance. Users can also customize how information is displayed and managed, supporting flexible and detailed analysis.

Real-Time Mode: Enable continuous live tracking of parameters without the need to refresh the chart.

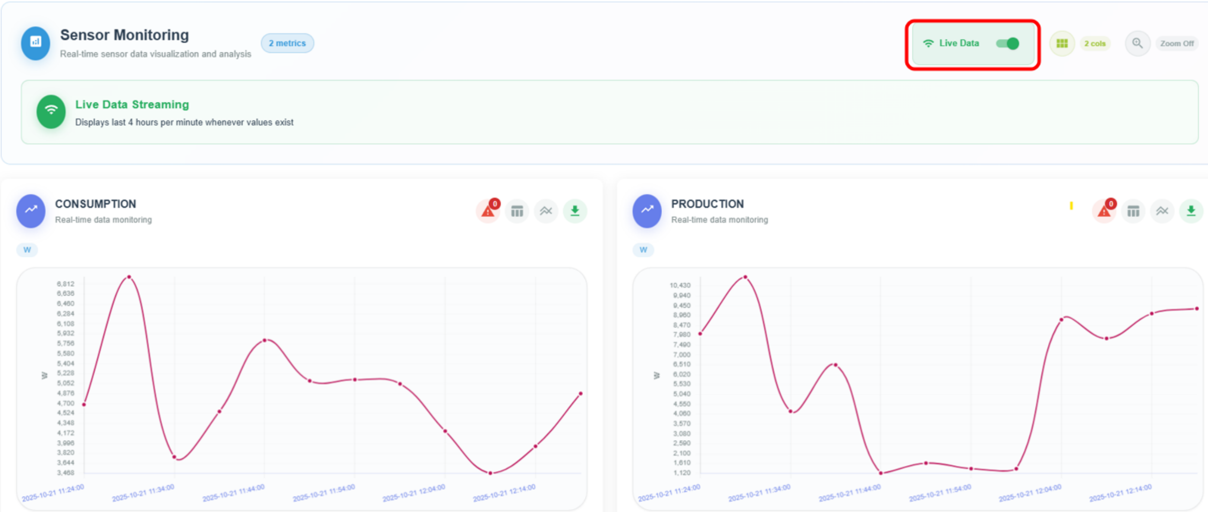

Figure 16 Track PM10 levels in real-time.

Custom Range Selection: Define a “from-to” range to focus on a particular period of interest, with the option to adjust the level of detail (e.g., minutes, hours, days).

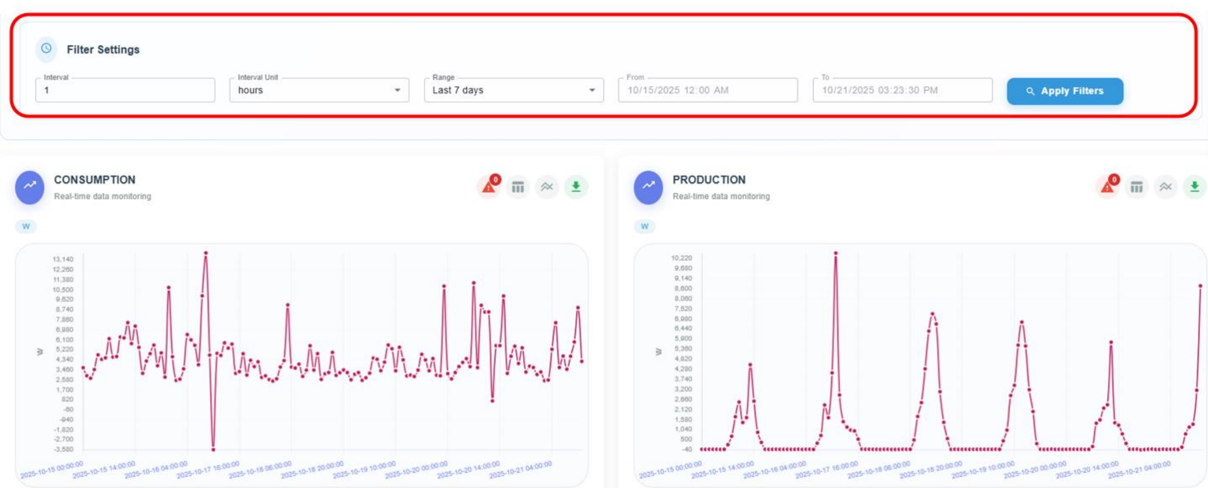

Figure 17. Track PM10 levels for a using a custom time interval selection.

Multi-Source Comparison: View and compare data from multiple measurement points (e.g., different rooms within the same building) on the same chart.

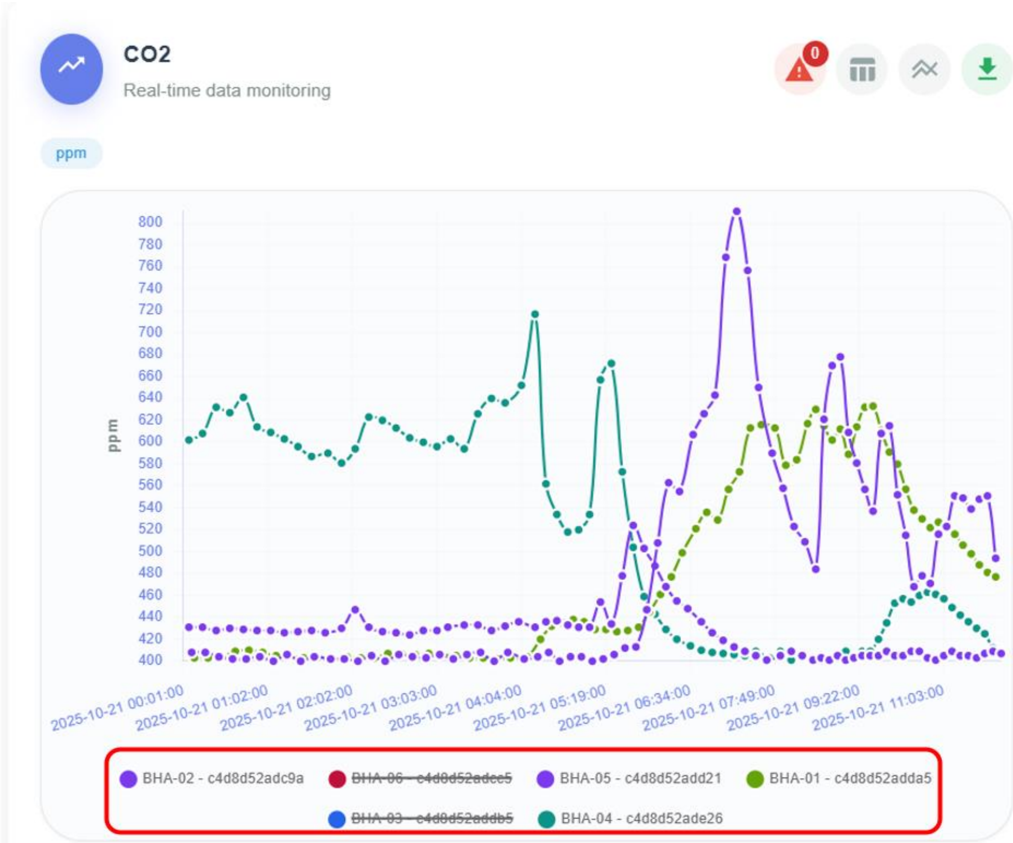

Figure 18. Track PM10 levels in selected rooms within the building.

Data Export: Download measurement data for further analysis or reporting in external tools.

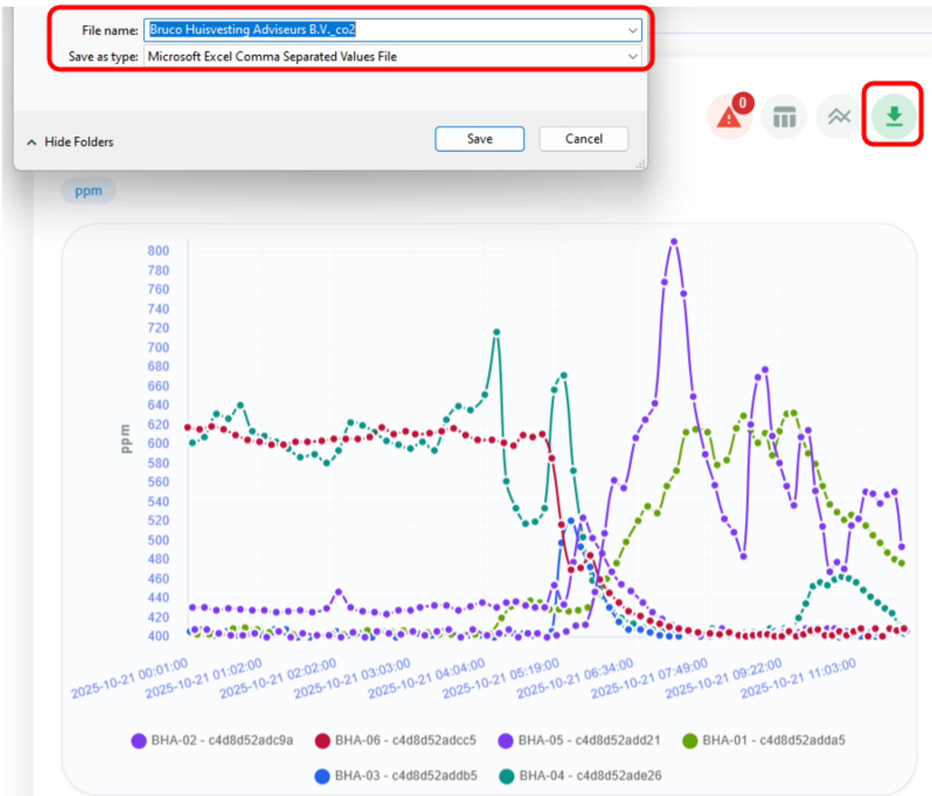

Figure 19. Download PM10 measurements in CSV format.

6.4 Advanced Features

In addition to core monitoring and visualization, the platform offers advanced features that improve efficiency and support strategic management. These capabilities include real-time alarms, map-based visualizations, comprehensive sensor metadata, weather integration, predictive analytics, and decision support tools. Together, they help building operators make proactive, data-driven decisions. The dashboard's advanced features can be detailed as follows:

6.4.1 Map-Based Visualizations

The dashboard shows the building's location on a map, providing geographic context for monitoring and management. While individual sensors are not displayed, operators can better understand the building's surroundings and external factors, such as nearby infrastructure or weather conditions.

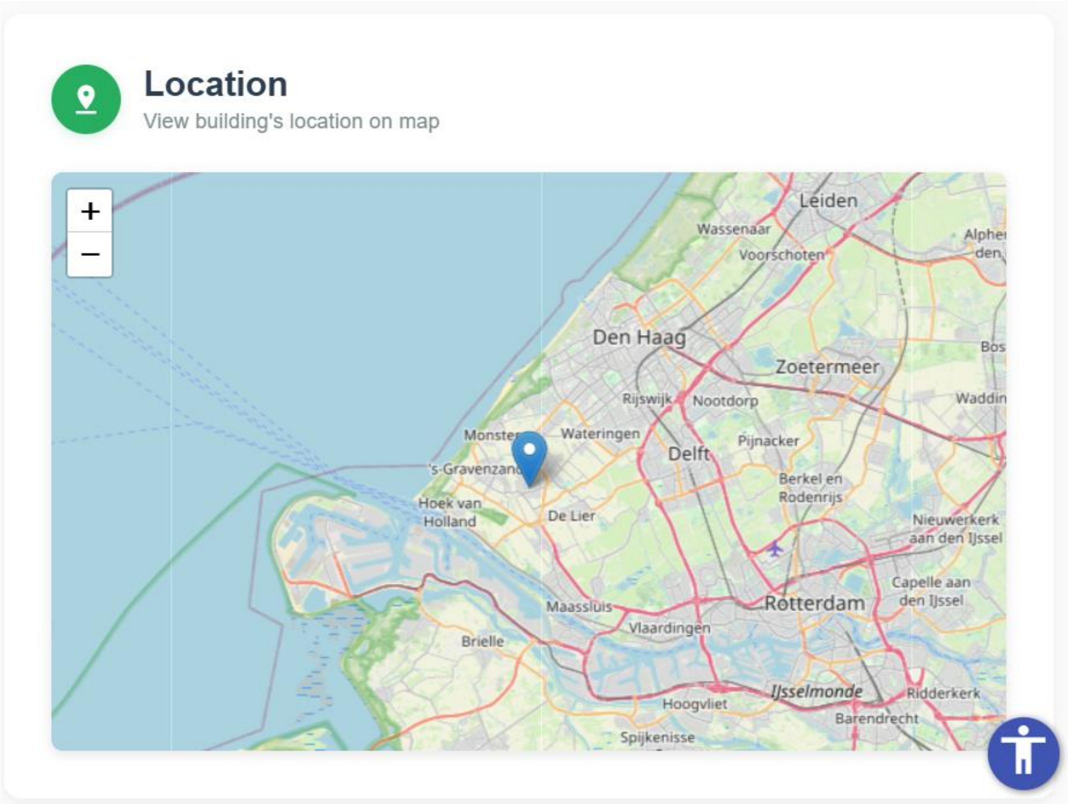

Figure 20. Building location

6.4.2 3D Visualisation

Through the uploaded IFC file, the facility manager gains access to a 3D visualization of the building, allowing interactive exploration and deeper insight into its structural and spatial characteristics. The user has access to this feature through the Dashboard of BOMS.

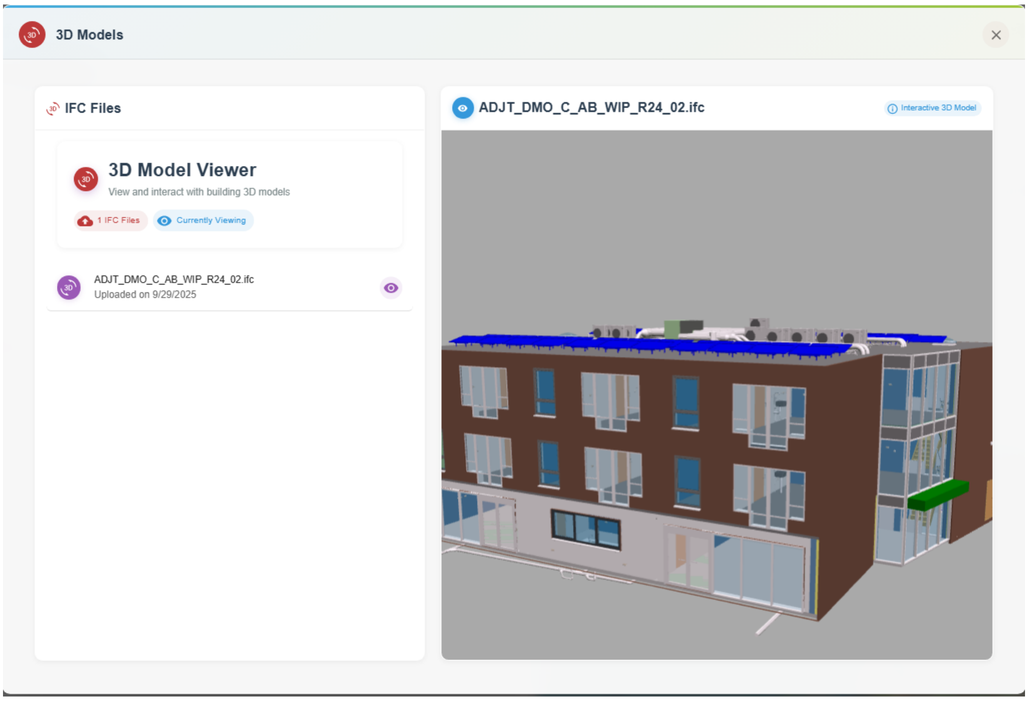

Figure 21. 3D Model Viewer of a building

6.4.3 Structure Tree of the Spaces

BOMS automatically detects all building spaces and visualizes them in a hierarchical tree structure, integrating data about sensors and subspaces. User can navigate the entire building or focus on a specific space, exploring its spatial layout and viewing all associated monitoring data. This visualization makes it easy to analyze sensor distribution and understand the organization of the facility.

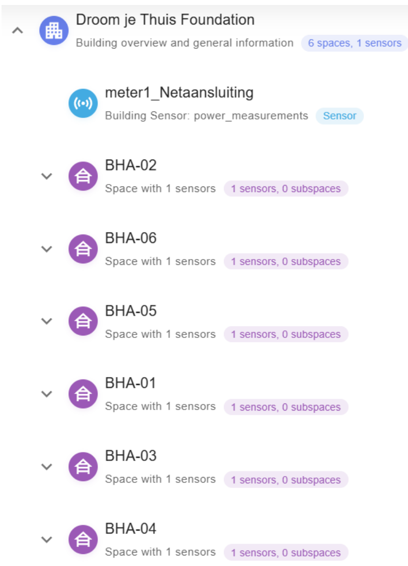

Figure 22. Hierarchical Visualization of Building Spaces and Sensor Integration in BOMS

6.4.4 Weather Integration

The dashboard uses weather data from Open-Meteo[19] to provide reliable forecasts that enhance building performance monitoring. By combining real-time building data with external weather information, users can see how outdoor conditions, such as temperature, humidity, and solar radiation that impact the indoor environments. This improves data interpretation and supports more informed decisions.

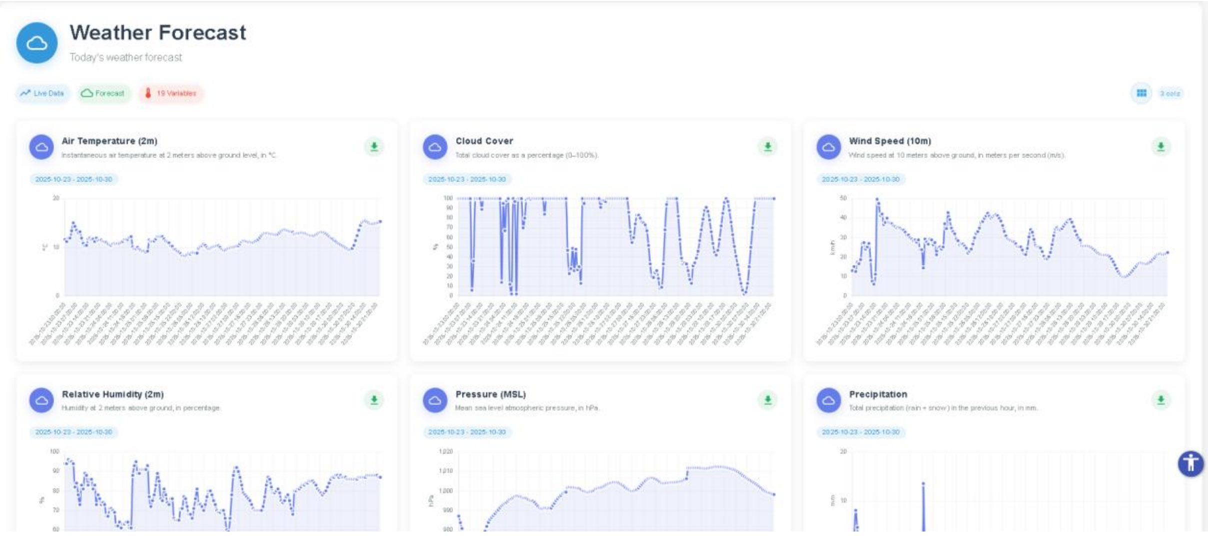

Figure 23. Weather Forecast Page

6.4.5 Sensors Forecast

The Sensor Forecasts page provides interactive, predictive insights. Users can filter by sensor, metric, and date range. Key metrics such as Forecast Accuracy, Data Points, Trend Direction, and Confidence Range are displayed in simple cards.

The main chart overlays actual values with forecasted consumption, showing P10 (lower bound), P50 (median), and P90 (upper bound) confidence intervals. This allows operators to quickly identify trends, anomalies, and performance issues, supporting efficient monitoring and decision-making.

[19] https://open-meteo.com/

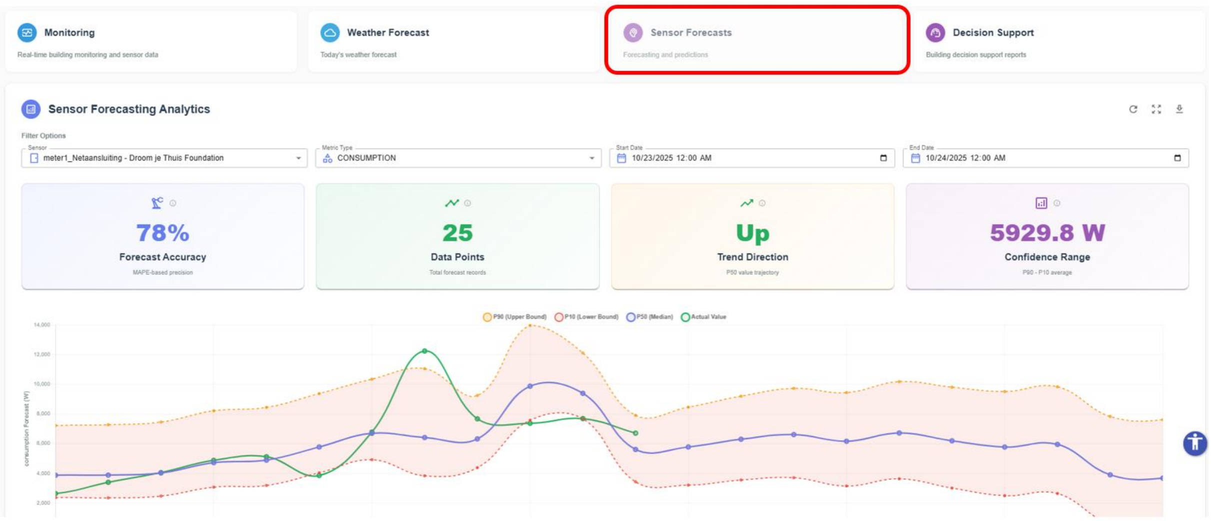

Figure 24. Sensors Forecast Page

6.4.6 Decision Support

The Decision Support Tool integrates directly with the dashboard, providing building facilitators with real-time insights and actionable recommendations in a single interface. It collects sensor data, forecasts, and external signals like weather and energy prices, then analyses them through agents for comfort, energy efficiency, and anomaly detection. The dashboard displays the results as intuitive reports, highlighting current conditions, potential issues, and recommended actions, enabling proactive, data-driven decision-making.

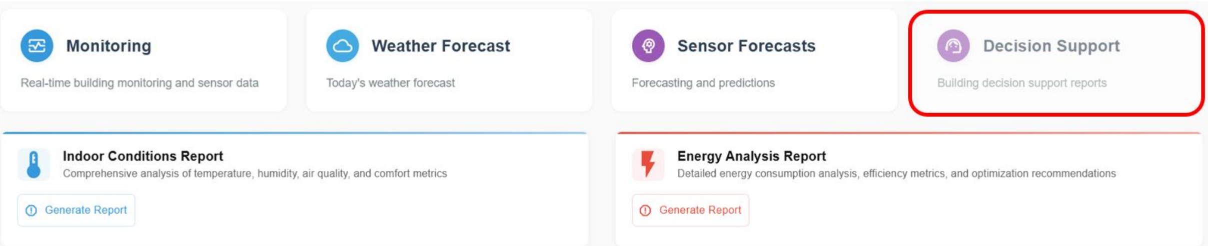

Figure 25. Decision Support Page

6.4.7 Alarms / Notifications

The system monitors critical parameters in real-time and triggers alerts when thresholds set by the building operator are crossed. This ensures that operators are immediately aware of issues, minimizing downtime and preventing unexpected events. For example, if a specific air quality measure exceeds the defined limit, an alert is sent to the operator to enable prompt preventive action.

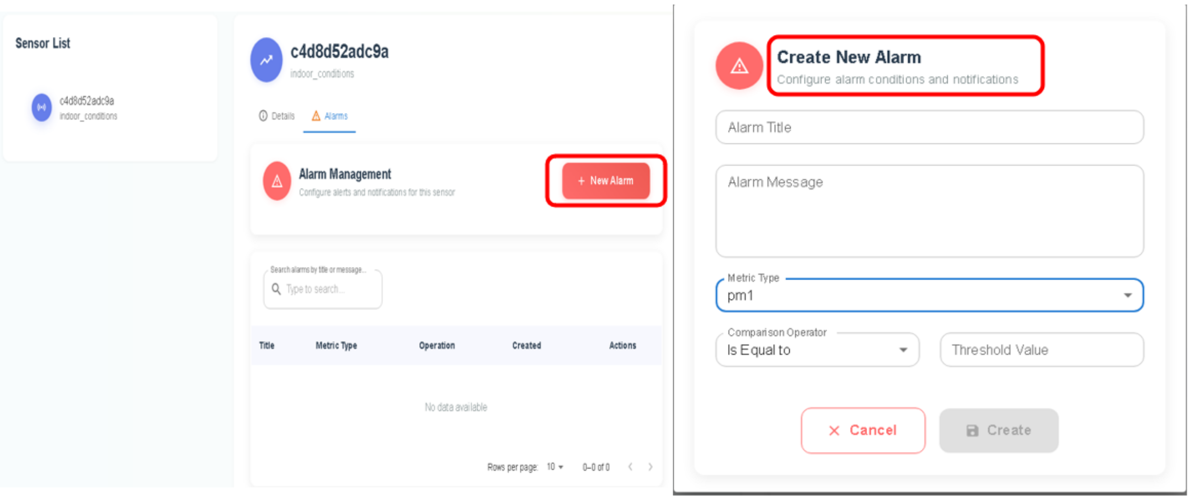

Figure 26. Alarm and notification settings

7 RE GIS

7.1 Introduction of the GIS

As part of T6.1, RE GIS was extended with a new accessibility scoring methodology that allows users to evaluate and record accessibility factors for any selected building. Buildings as objects can be generated by users who have login credentials, and through the GIS interface, users (no login required) can select a building from the previously made objects on the map and access a building profile containing accessibility attributes, user-provided scores, and contextual geospatial information.

RE GIS builds upon and extends the RE Suite of DMO which is a software Solution for collecting, structuring, analysing and disclosing real estate information. The term 'RE' stands for Real Estate, encompassing both building and civil infrastructure (in total aka the built environment), while 'Suite' represents a collection of software components, providing a cohesive and integrated solution. The RE Suite encompasses a range of applications designed to support various aspects of real estate management life cycle.

RE GIS extends the RE Suite with geospatial intelligence by integrating real estate data into 2D and 3D GIS environments, enabling location-based analysis and visualization of assets within their spatial and environmental context. It provides optical detection and intuitive exploration of properties in high-quality 3D maps, allowing users to identify spatial relationships, patterns, and risks across portfolios, districts, and cities. RE GIS supports multiple Levels of Detail (LoD) per property, ranging from high-level overviews such as location, footprint, and zoning to detailed building, component, and condition information, aligned with standards such as GIS, CityGML, and BIM.

Under this context, the RE GIS module supports the AccesS BOMS and AccesS Suite by enabling spatially explicit evaluation of building accessibility. The GIS environment allows users to select buildings on the map, review their accessibility characteristics, and contribute new assessments through a structured scoring system. The methodology implemented in RE GIS takes into consideration the principles and scoring structure identified within WP3, which lays the foundation for the AccesS Accessibility Assessment Scheme, currently under development within WP8. The aim is to operationalize its principles in a digital, user-friendly form based on user-generated feedback and scoring user experience.

7.2 Conceptual Basis

The methodology builds upon the three analytical pillars defined in the Accessibility and Inclusiveness Certification Model as part of the AccesS Assessment Scheme, described in D3.1 Review of Existing Certification Schemes and Integration of stakeholder insights:

1. **Disability Types (DT)**: Categories of user needs and disabilities (e.g., mobility, visual, auditory, cognitive).
2. **Assessment Dimensions (AD)**: Thematic areas in which accessibility performance is assessed (e.g., circulation, signage, interaction with building systems).
3. **Evaluation Criteria (EC)**: Concrete, measurable aspects of accessibility to be scored by assessors.

These pillars are structured hierarchically and linked through weighting and scoring rules, ensuring consistency, comparability across buildings, and the aggregation of results into composite accessibility indicators.

This methodological framework has been used as a reference basis for defining both the data collection approach and the scoring logic implemented in the RE GIS tool. However, given that the RE GIS tool is designed to support large-scale deployment and crowdsourced data collection, the full expert-based Evaluation Criteria have been intentionally simplified in the first version of the tool.

Specifically, the initial implementation focuses primarily on Pillar 3 (Evaluation Criteria), while selectively integrating elements from the Disability Types and Assessment Dimensions pillars. This approach allows the formulation of concrete, easy-to-understand questions that can be reliably answered by non-expert users, while remaining aligned with the underlying accessibility assessment logic.

Within this context, and with the objectives of increasing spatial coverage, maintaining up-to-date information, and capturing real user experiences, the RE GIS tool enables crowdsourced accessibility assessments through a set of targeted evaluation questions. These include, for example:

- usability of ramp gradients and landing widths
- availability and placement of tactile surfaces
- clarity of voice-feedback systems
- acoustic comfort in circulation areas
- accessibility of smart HVAC or lighting controls
- clarity of emergency evacuation signage

It should be noted that crowdsourced inputs are intended to complement, rather than replace, expert-based diagnostics. They enable continuous monitoring of accessibility performance over time and provide valuable evidence for identifying priority areas for detailed expert evaluation. In future development phases, the integration of structured expert assessments and advanced weighting mechanisms will be further investigated to enhance analytical depth and certification alignment.

7.3 Implementation in RE GIS

RE GIS will be coupled with BOMS platform via a dedicated link that redirects users to the RE GIS environment. Any user with a registered account can create objects, while login is not required specifically for the creation of scores. The resulting accessibility ratings is configured to be publicly available, enabling transparency and broad access to the assessment results while maintaining controlled user authentication for data input.

The development of the RE GIS tool focused on integrating the generated building profile data and crowdsourced-based accessibility assessment directly into an intuitive, map-based environment. Figure 27 illustrates the enhanced GIS interface, where buildings can be selected on a map and the building profile provides the general information of the building.

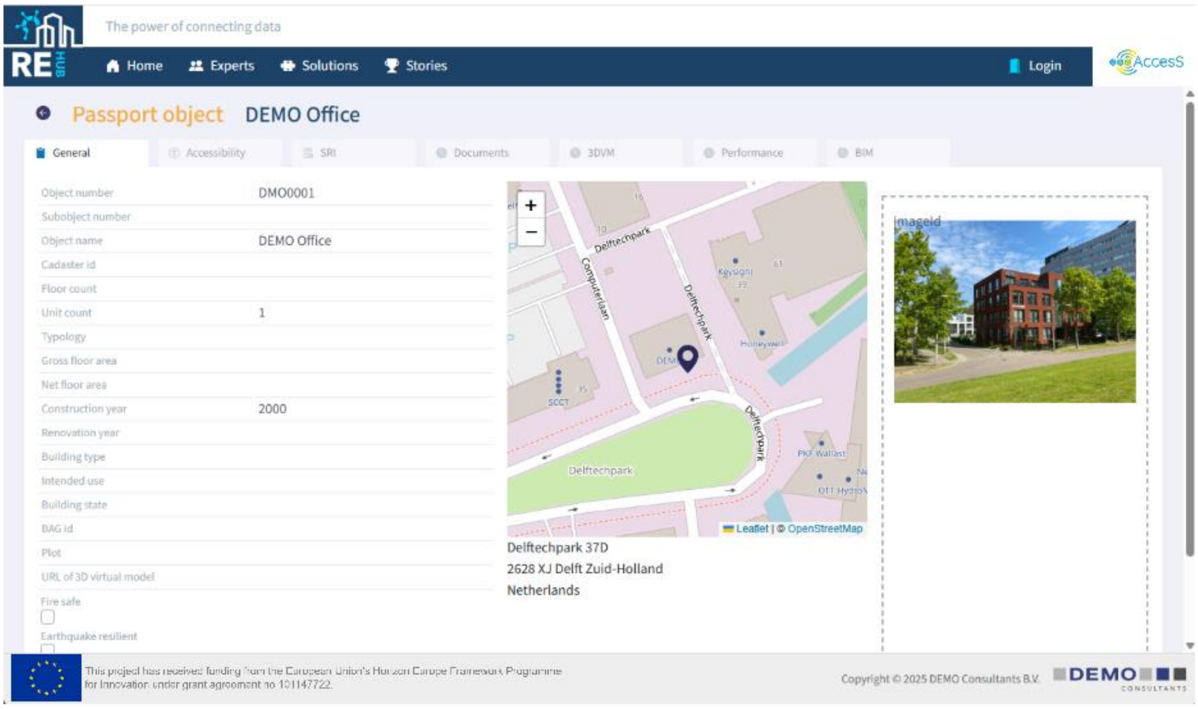

Figure 27 RE GIS Building Profile

By integrating the accessibility score methodology through an interactive scoring module (Figure 28), users can assign scores, mark criteria as not applicable, and submit partial assessments, all of which are stored and dynamically aggregated into each building's accessibility profile. Together, these features demonstrate the successful operationalisation of accessibility featured within RE GIS, enabling continuous, user-driven updates and spatially contextualised insights into building inclusiveness.

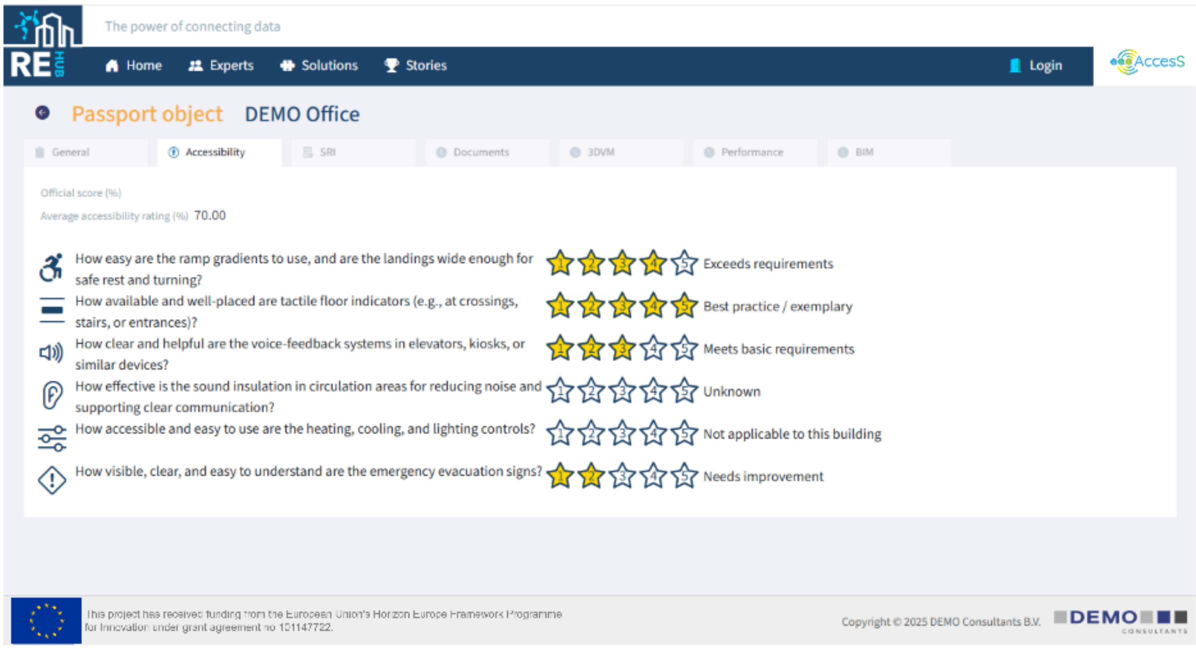

Figure 28 RE GIS Accessibility

8 Conclusions

Deliverable D6.1 encompasses the full development of T6.1 concerning BOMS. Within BOMS, a dashboard has been developed to monitor all relevant measurements across the pilot sites. Building on this, the deliverable presents the two main sub-modules of BOMS: the decision support system (DSS) and the forecasting engine. The DSS uses intelligent agents to provide actionable recommendations for building operations, helping to improve energy efficiency, maintain occupant comfort, and enhance system flexibility. The forecasting engine, based on Chronos models, uses AI to predict energy consumption, renewable energy production, and indoor conditions, providing essential input to the DSS.

Key achievements of this deliverable include:

Monitoring dashboard: A platform to visualize and track all pilot measurements in real time.

Database setup and data processing: A TimescaleDB database was set up to store and manage all pilot data. A dedicated data model was designed to ensure structured and scalable data organization, while an API was developed to enable data exchange between BOMS components and external systems.

Forecasting engine (Chronos): AI-driven models that accurately predict energy use, renewable generation, and indoor conditions to support the DSS.

Decision support system (DSS): An agent-based module that translates data and forecasts into practical recommendations for building operation and optimization.

Analytics and visualization: Dashboards and visual tools that make data insights easy to understand and use. In addition to standard monitoring plots, advanced features have been added for the pilot sites, such as a BIM viewer for visualizing buildings and a map-based interface showing their locations.

Location-specific management of accessibility features: The RE GIS tool transforms accessibility assessment into a practical scoring with user-generated feedback, and describes the visualisation of building-level accessibility information on the map.

In summary, D6.1 demonstrates that combining real-time monitoring, predictive forecasting through Chronos, and agent-based decision support enables smarter, more efficient, and flexible building management. The next steps will focus on refining Chronos' forecasting models, enhancing DSS optimization strategies, and continuing validation in the pilot sites. Future work will also include integrating all available data from the pilot sites, retraining the models with larger datasets, and improving their accuracy and adaptability. Also exchange of data with BOMS and RE GIS through API or import will be considered as well as integration of accessibility related data (such as wheelchair friendly buildings) from external databases in the RE GIS component. These developments will take place during the integration and testing tasks of WP7 as well as demonstration and evaluation activities of WP10-WP12 and will support BOMS evolution into a highly adaptive, user-friendly tool for sustainable building management, coupled with GIS.

9 References

[1] G. E. Box, G. M. Jenkins, G. C. Reinsel and G. M. Ljung, Time Series Analysis: Forecasting and Control, John Wiley & Sons, 2015.

[2] R. J. Hyndman and G. Athanasopoulos, Forecasting: Principles and Practice, OTexts, 2018.

[3] O. Shchur, A. C. Turkmen, N. Erickson, H. Shen, A. Shirkov, T. Hu and B. Wang, "AutoGluon–TimeSeries: AutoML for Probabilistic Time Series Forecasting," in *Proceedings of the Second International Conference on Automated Machine Learning* , 2023.

[4] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[5] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting,* pp. 679-688, 2006.

[6] M. M. T. H. K. Y. a. M. K. T. Sawada, "Office-in-the-Loop: an investigation into Agentic AI for advanced building HVAC control systems," *Data-Centric Engineering,* vol. 6, 2025.

[7] J. S. J. K. S. C. J. H. S. M. H. S. Y. L. J. L. J. Y. G. L. M. S. S. Y. Jeyoon Lee, "Agentic Built Environments: a review," *Energy and Buildings,* vol. 346, 2025.

[8] S. M. L. M. P. M. Muhammad Arslan, "Monitoring indoor environmental conditions in office buildings using a sustainable Agentic RAG-LLM system," *Energy and Buildings,* vol. 347, 2025.

[9] S. Y. Jaemin Hwang, "AI agent-based indoor environmental informatics: Concept, methodology, and case study," *Building and Environment,* vol. 277, 2025.

[10] C. Benavente-Peces, "Energy efficiency in the next generation of smart buildings—Supporting technologies and techniques," *Energies,* vol. 12, p. 4399, 2019.

Enhancing Accessibility and Sustainability in Smart Cities and Smart Buildings: The Universal Accessibility Suite Initiative

www.euproject-access.eu/en/

@EU project AccesS

@EUprojectAccesS

@EUprojectAccesS

This project has received funding from the European Union's Horizon Europe research and innovation programme under the grant agreement number 101147722. The European Union is not liable for any use that may be made of the information contained in this document, which is merely representing the authors' view.