Ref. Ares(2025)3516144 - 30/04/2025

atelier
Positive Energy Districts

AmsTErdam BiLbao cItizen drivEn smaRt cities

Deliverable 9.3

Web-based online indicator observatory

WP9, Task 9.3

Date of document

30/04/2025 (M 66)

Deliverable Version:	D9.3, V 1.1
Dissemination Level:	PU
Author(s):	UDEUSTO (Roberto Garay, Ainhoa Arriazu-Ramos, Itziar Salaberria, Xabier Cantero, Aitor Díez, Ane Oleaga) CARTIF (Jose Luis Hernandez)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No.864374

List of beneficiaries

No	Name	Short name	Country
1	GEMEENTE AMSTERDAM	AMST	Netherlands
2	AYUNTAMIENTO DE BILBAO	City of Bilbao	Spain
3	FUNDACION TECNALIA RESEARCH & INNOVATION	Tecnalia	Spain
4	NEDERLANDSE ORGANISATIE VOOR TOEGEPAST NATUURWETENSCHAPPELIJK ONDERZOEK TNO	TNO	Netherlands
5	FUNDACION CARTIF	CARTIF	Spain
6	STICHTING WAAG SOCIETY	Waag Society	Netherlands
7	STICHTING HOGESCHOOL VAN AMSTERDAM	AUAS	Netherlands
8	PAUL SCHERRER INSTITUT	PSI	Switzerland
9	STEINBEIS INNOVATION GGMBH	SEZ	Germany
10	BUDAPEST FOVAROS ONKORMANYZATA	MunBud	Hungary
11	MUNICIPIO DE MATOSINHOS	Matosinhos	Portugal
12	RIGA MUNICIPAL AGENCY "RIGA ENERGY AGENCY"	Riga EnAg	Latvia
13	KOBENHAVNS KOMMUNE	COP	Denmark
14	HLAVNE MESTO SLOVENSKEJ REPUBLIKY BRATISLAVA	BRATISLAVA City	Slovakia
15	GMINA MIEJSKA KRAKOW - MIASTO NA PRAWACH POWIATU	City of Krakow	Poland
16	UNIVERSIDAD DE LA IGLESIA DE DEUSTO ENTIDAD RELIGIOSA	UDEUSTO	Spain
17	CLUSTER DE ENERGIA	CEPV	Spain
18	IBERDROLA ESPANA SA	IBE	Spain
19	TELUR GEOTERMIA Y AGUA SA	TELUR	Spain
20	ENTE VASCO DE LA ENERGIA	EVE	Spain
21	SPECTRAL ENTERPRISE BV	Spectral	Netherlands
22	MAANZAAD BV	Republica	Netherlands
23	EDWIN OOSTMEIJER PROJEKTONTWIKKELING BV	EdwinOostmeijer	Netherlands
24	STICHTING AMSTERDAM INSTITUTE FOR ADVANCED METROPOLITAN SOLUTIONS (AMS)	AMS Institute	Netherlands
25	STICHTING WATERNET	WATNL	Netherlands
26	DNV NETHERLANDS B.V.	DNV	Netherlands
27	GROENE ENERGIE ADMINISTRATIE BV	Greenchoice	Netherlands
28	ZABALA INNOVATION CONSULTING. S.A.	ZABALA	Spain
29	FRAUNHOFER GESELLSCHAFT ZUR FOERDERUNG DER ANGEWANDTEN FORSCHUNG E.V.	Fraunhofer	Germany

Document History

Project Acronym	ATELIER		
Project Title	AmsTErdam and BiLbao cItizen drivEn smaRt cities		
Project Coordinator	City of Amsterdam atelier.eu@amsterdam.nl		
Project Duration	01/11/2019 – 30/04/2026 (78 Months)		
Deliverable No.	D9.3: Web-based online indicator observatory		
Diss. Level	Public (PU)		
Deliverable Lead	UDEUSTO		
Status		Working	
		Verified by other WPs	
	x	Final version	
Due date	30/04/2025		
Submission date	30/04/2025		
Work Package	WP 9 - Monitoring, impact assessment and evaluation		
Work Package Lead	PSI		
Contributing beneficiary(ies)	UDESUTO, CARTIF		
DoA	Impacts will be assessed by comparing the performance indicators in reference alternatives (BAU from WP 2) and the actual performance indicators in Task 9.2. The impact assessment will comprise indicators covering the traditional sustainability dimensions, including environmental, economic and social aspects. For environmental aspects, not only the operation/use phase, but also the construction and end-of-use disposal and recycling phase are covered from a life cycle perspective. This is essential because energy efficiency measures can lead to an increase of upstream/downstream materials and energy for manufacturing and/or recycling before/after the operation of the buildings. Evidence shows that the dominant life cycle environmental impacts are gradually shifted from operation to the construction of buildings. It is therefore important to apply Life Cycle Assessment (LCA) to assess any PED measure of interest, in order to effectively avoid burden shifting from PED to elsewhere in the city or beyond. The environmental and economic aspect of the impact assessment will focus on the resulting life cycle costs of the PED measures. The social aspect will be assessed mostly based on a stakeholder survey or interviews and other data monitored provided by WP7. In addition, given the available data provided by the PED demonstration sites, impacts will be also assessed in energy, mobility and spatial aspects, such as percentage of renewable energy integrated, walkability of districts, connectivity between facilities and infrastructures, flexibility through a multi-modal strategy in mobility, convenience, etc.		

Date	Version	Author	Comment
26/03/2025	0.1	RGARAY	First draft of deliverable

	0.2	JLHERNANDEZ, AARRIAZU, ISALABERRIA, XCANTERO	Contributions
04/2025	0.9	RGARAY	Full version
04/2025	1.0	AOLEAGA, RGARAY	Revision & Editing
28/04/2025	1.1	RGARAY	Modifications after internal review

Table of Contents

List of Figures

List of Tables

1 Introduction

ATELIER is a research project on Positive Energy District (PED) development. Within ATELIER, two such districts will be developed in Amsterdam and Bilbao. ATELIER WP9 aims at performing the monitoring of such developments.

This report is a deliverable (D9.3) that covers the development of a Web-based online indicator observatory, that is updated on a regular basis with information on Key Performance Indicators (KPIs) gathered for the demonstration sites.

This web-based observatory is built based on several processes and items, as follows:

1. Data gathering & consolidation. Data is obtained from individual data sources, integrated and consolidated into a single dataset
2. Data is made available online for automated reading by authorized users through an Application Programming Interface (API)
3. KPIs are calculated
4. A visualization layer is developed and made available online through an open website.

Data gathering and consolidation is performed once per month, about 1 week after the end of each reporting period. This is due to limitations in some of the data sources[1].

Data consolidation, KPI calculation and visualization processes are made consistent with the methodology outlined in D9.2. Specific scenarios are defined so that the performance of the PEDs is corrected for climate and other specific issues so that it is traceable to the expectations during the design phase.

The data structure has been defined commonly for Amsterdam and Bilbao, but individual databases are deployed for each of the sites.

The process presented in this report has been tested with monitoring data for Amsterdam, particularly with data flows for the Republica site. A similar approach will be used for replication in Poppies and Bilbao, along the population of the districts, and the setting-up of data flows.

Interaction with the database is possible at two independent layers. Consolidated data is made available for secondary research/use through an Application Programming Interface (API, i.e. for other researchers and WPs within ATELIER), as well as through a web interface for final use (i.e. for direct assessment of the PEDs, within or outside the ATELIER consortium).

[1] Several data sources are linked to invoicing processes. Although data is reported with down to hourly resolution, it is only gathered once per month and requires some pre-processing before delivery.

2 Basic criteria

ATELIER D9.2 sets the ground for monitoring activity. In this section, a brief summary of this document is presented, along with some specificities regarding data delivery.

2.1 Considered scenarios

In the frame of ATELIER, PED performance is compared with the performance expectations resulting from the design process[2]. It should be considered that these expectations are based on assumptions with regards to climate, building usage, and import/export balances with the surrounding energy system. A set of 6 intermediate scenarios are defined to fairly confront the design (scenario 1) and monitored (scenario 4) PED. This is summarized in the following figure:

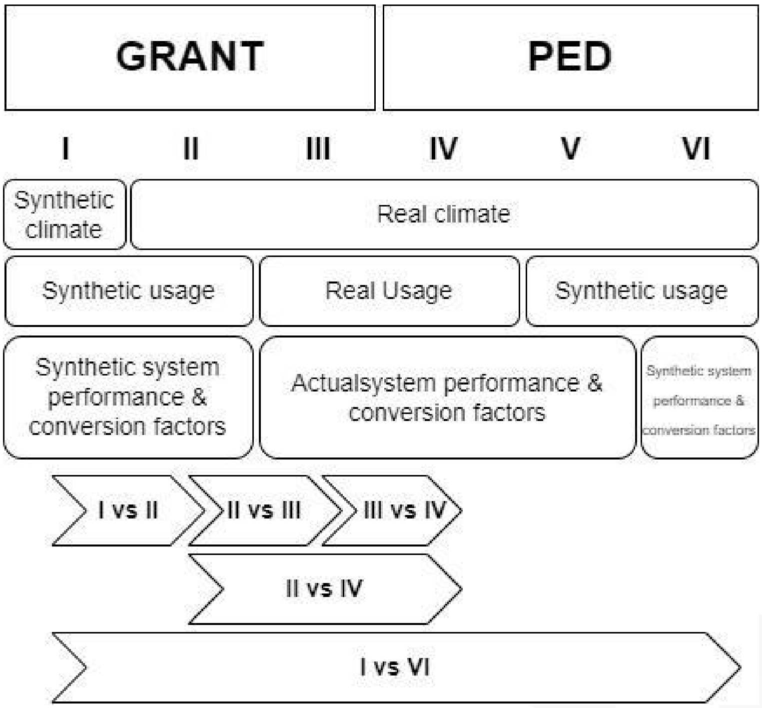

Figure 1 Synthesis of Scenarios and confrontations (Source: D9.2)

[2] These were reflected in the Grant Agreement (BEST tables)

2.2 Climate data

Climate data is sourced from OpenMeteo[3].

2.3 Aggregation at building & block levels, and Filling of gaps

Considering the scope of assessment at PED level, as well as data protection issues, data is aggregated at building and/or block level. This ensures the privacy of any disclosed information. Also, it allows to oversee particular performance and focus on the urban scale.

Aggregation processes require that data is available for all individual meters to be aggregated. Although meters are robust systems, there are short periods of unavailability, as well as cases where some of the meters are installed later than others. For this reason, specific provisions have been taken to fill the gaps and make aggregation processes viable.

2.4 Update frequency

Data flows into the observatory are heterogeneous. Although some of them are automated flows (i.e. API connections or automated FTP dumps), there are many manual deliveries (i.e. e-mail deliveries at the end of the month).

It should be considered that the observatory is meant to deliver a consistent set of energy balances. This requires that information from all data sources is processed correctly.

In agreement with this, data update frequency is limited to the frequency of manual deliveries. Every month, data flowing from all sources are integrated into the database and consolidated. Then automated processes are executed to update KPIs and visuals. This process is expected to be updated 2 weeks after the end of the reported period in each data delivery[4].

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

[4] Some data providers require some time to perform their own internal prosses prior to delivering the data.

3 Architecture for data processing and KPI calculation

In order to enable efficient monitoring and decision-making across different locations (Amsterdam and Bilbao), a robust architecture has been established to handle data acquisition, processing, and KPI calculation. The system has been designed to integrate data from multiple sources, process this data in a consistent manner, and make the results accessible through user-friendly dashboards.

As illustrated in the diagram in Figure 1, the architecture is considering the data from various local sources and stored in two distinct databases, one for Amsterdam and another for Bilbao. This architecture relies on these data sources and the description of the acquisition, formats and communication is not part of this section.

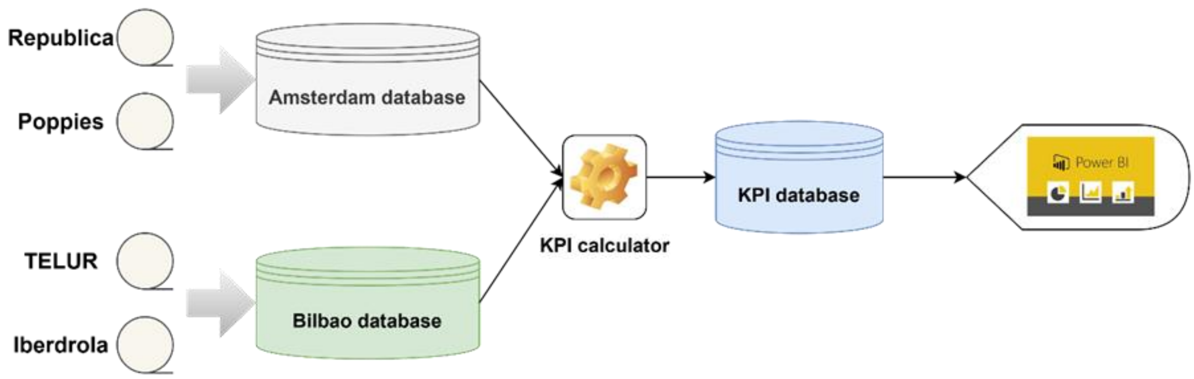

Figure 2 Simplified architecture for the KPI calculation

To ensure consistency and reliability in performance monitoring, a dedicated KPI calculator module has been developed. This component accesses data from both the Amsterdam and Bilbao databases (see sections 4 and 5) and performs standardized calculations to generate a unified set of KPIs. These indicators are then stored in a centralized KPI database, which serves as the single source of consolidated performance metrics across the system.

Finally, the calculated KPIs are visualized using Power BI, allowing stakeholders to explore the data through dynamic dashboards and gain actionable insights into system performance and efficiency. This layered architecture not only ensures data integrity and scalability but also facilitates cross-site comparability and real-time analysis.

To provide greater transparency and operational clarity, the detailed architecture shown in Figure 2 outlines the step-by-step flow of data and the subsequent transformation into performance insights. This implementation ensures a traceable, modular, and scalable system that connects heterogeneous data sources with business intelligence tools.

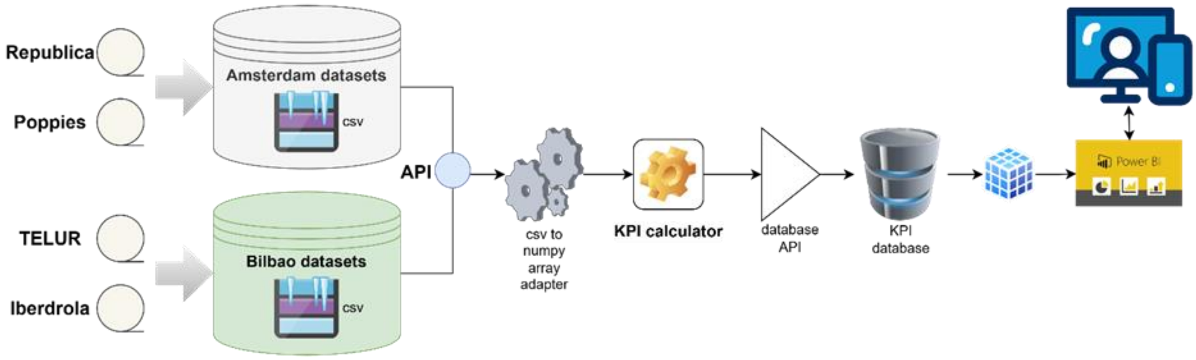

Figure 3. Detailed architecture for the KPI calculation

Data collection from the Amsterdam and Bilbao databases is performed via an API, responsible for translating the CSV files from these local repositories into a JSON (JavaScript Object Notation) object (see section 6 for more details). This object is easily translated into a DataFrame, which needs to follow the format detailed in Table 1. The KPI calculator requires a numpy array; therefore, there is a component in charge of accommodating the DataFrame into numpy to assure data inputs. This engine uses structured datasets from both sites and applies a series of transformations.

Table 1. Format for the DataFrame used in the KPI calculator

Timestamp	Building ID-1	…	Building ID-N
yyyy-mm-dd hh:MI:ss	value1-1		value1-N
yyyy-mm-dd hh:MI:ss	value2-1		value2-N

The centralized KPI calculator is storing data and applies a series of aggregations and formulae to compute relevant Key Performance Indicators (see section 7). The output of the KPI calculator is written to a central KPI database, which acts as a unified repository for all calculated metrics. This centralization is crucial for enabling cross-site performance analysis and ensures that downstream systems always work with the most up-to-date and validated indicators. In this sense, the storage is rendered through a KPI API, enabling both the creation of the database structure, initial insertion of data and continuous storage of KPIs.

The design allows for the KPI logic to be maintained and versioned independently, facilitating updates without altering the original datasets. It also enables the harmonization of data formats across locations, ensuring that calculated KPIs are consistent and comparable.

Finally, the PowerBI engine is used to query the database, extract the KPIs and provide some additional aggregations in the form of a data cube structure to support advanced querying and multi-dimensional analysis. This facilitates slicing and dicing the KPIs by time, location, category, or any other analytical dimension, vastly enhancing the insight extraction capabilities. This allows KPIs to be exposed through a Power BI dashboard (see section 8). These dashboards are designed to be fully responsive, enabling access across devices—from desktops to mobile platforms thanks to being embedded into a Web page.

This architecture not only guarantees robustness and scalability but also aligns with best practices in industrial data analytics, offering a strong foundation for continuous improvement, benchmarking, and decision support.

4 Data flows

Data from different data suppliers is aggregated in the final dataset. The following data sources, formats and delivery frequency are considered:

- Desing data.
  - Climate: Provided by Meteonorm[5] (licensed software). Typical meteorological year with hourly resolution
  - PED performance: Design values established in the BEST tables within the ATELIER Grant Agreement (& succeeding Amendments)[6]. Yearly resolution.
- Climate data:
  - OpenMeteo[7]. Publicly available. Hourly resolution. Queried on-demand through the web-interface.
- Amsterdam
  - Electricity data. Provided by Spectral. Hourly or sub-hourly resolution. Monthly deliveries by e-mail.
  - Heat data. Provided by Spectral (originally metered by Vaanster) Daily resolution. Monthly deliveries by e-mail. Delay of a few weeks after the end of the month.
- Bilbao
  - Electricity data for large buildings in Central and South areas. Provided by Iberdrola Clientes. Hourly or sub-hourly resolution. Daily delivery through an FTP server
  - Electricity data for residential buildings. Provided by Iberdrola Distribution. Space-aggregated[8]. Monthly deliveries by e-mail. Delay of a few weeks after the end of the month.
  - Heat & heat Pump data. Provided by TELUR. Hourly or sub-hourly resolution. Queried on demand through API and web-interface

[5] https://meteonorm.com/en/
[6] This information source is not made available to the public.
[7] https://open-meteo.com/
[8] To comply with GDPR regulation

5 Data consolidation

PED monitoring data collected from various data sources is consolidated in agreement with the following general criteria:

- File formatting and timestamps are made compatible
- Cumulated signals[9] are differentiated
- Missing data is incorporated by interpolation or the development of simplified models.
- Time aggregated data (i.e. at day or monthly aggregations) is re-distributed into hourly resolution. If possible, climate-based (i.e. HDD) processes are used for heating & cooling loads. For non-climate-related variables (i.e. electric loads for appliances), average values are calculated. All this is based on the provisions in D9.2.
- Data is integrated in the corresponding scenario (Scenarios 1 and 4) and then corrections are made to obtain scenarios 2, 3, 5 & 6. Again, considering the provisions in D9.2.

After this process, a dataset is obtained with the following data structure:

Table 2. Aggregation levels

Buildings	Format	SCNENARIO_NUMBER.PED_NAME.BLOCK_NAME.BUILDING.Variables
	Sample	SCN1.AMS.REP.BUI1.DEMAND.HEAT.SPACE
Block-level	Format	SCNENARIO_NUMBER.PED_NAME.BLOCK_NAME.CENTRAL.Variables
	Sample	SCN1.AMS.REP.HP.PROD.HEAT
Context	Format	SCNENARIO_NUMBER.PED_NAME.CONTEXT.Variables
	Sample	SCN1.AMS.CONTEXT.METEO.T

[9] This is a common way to measure energy in heat and electricity meters. There, the current load is added on top of the one in the previous hours.

Table 3. Specific variables

Building level	.DEMAND.HEAT.SPACE .DEMAND.HEAT.DHW .DEMAND.ELE.LIGHT .DEMAND.ELE.APPL .USE.HEAT.SPACE .USE.HEAT.DHW .USE.HEAT .USE.ELE.LIGHT .USE.ELE.APPL .USE.ELE.DHW .PROD.ELE.RENEW .USE.ELE.TOT .ELENET .PROD.HEAT.RENEW
Block level	.BATT.ELE.IN .BATT.ELE.OUT .BATT.ELENET .HP.PROD.HEAT .HP.USE.HEAT .EV.USE.ELE .HEAT .IMP.HEAT .EXP.HEAT .IMP.ELE .IMP.ELE.EXCL_PV .PROPV .HEAT.DH .EXP.ELE .EXP.ELE.EXCL_PV .ELENET .ELENET.EXCL_PV
Context information	.CONTEXT.METEO.T .CONTEXT.METEO.IS

6 REST API definition

In this section ATELIER's monitoring REST API[10] is specified. This API is designed to provide the data required for KPI calculations. A different API instance is deployed for each PED according to their particularities, providing the required endpoints to satisfy their own monitoring data requirements. In the same way, a different KPIs monitorization web is deployed for each PED.

Figure 4 shows the API documentation designed for the Amsterdam PED.

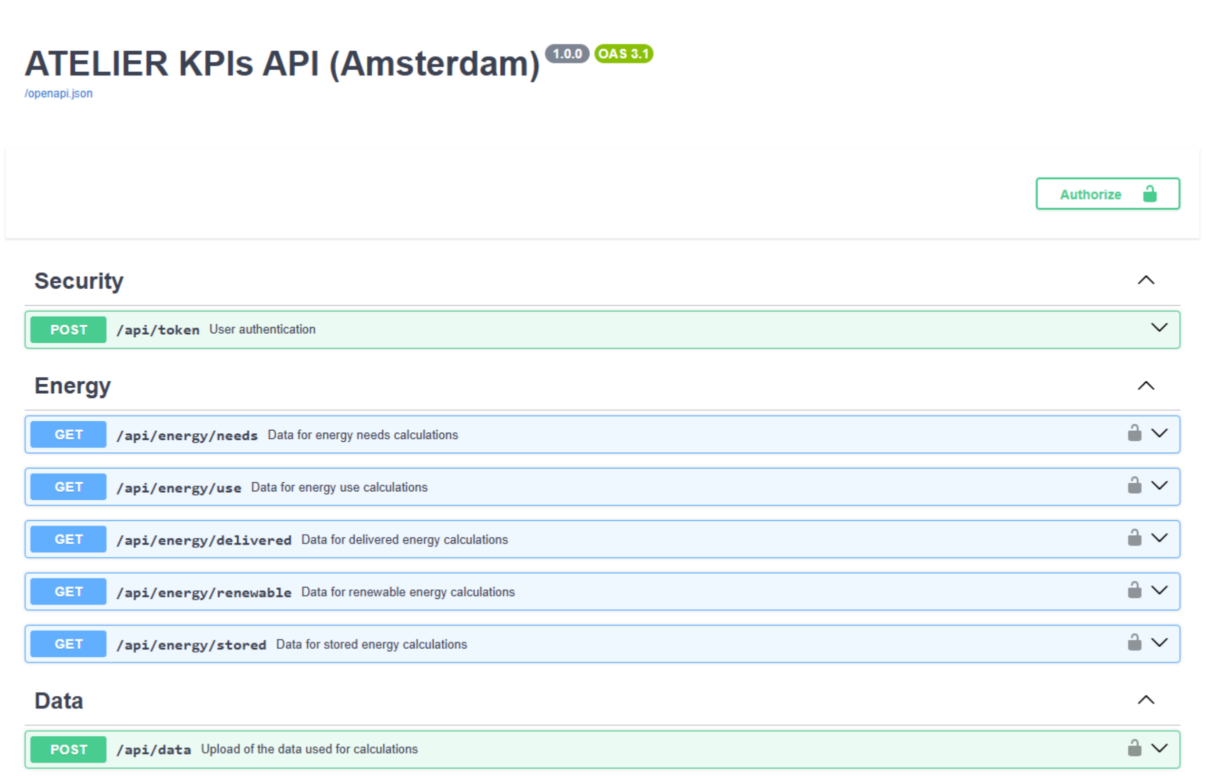

Figure 4. API documentation

As seen in the documentation, the API is divided into three sections. The first section focuses on API access security, implementing user authentication through JWT (JSON Web Token)[11] and Bearer Authentication[12]. Securing a REST API with a token-based authentication mechanism ensures that only authorized clients can access protected resources. The second section contains the endpoints related to the data required for calculating energy KPIs.

---

[10] A REST API is an application programming interface (API) that follows the design principles of the REST architectural style. REST is short for REpresentational State Transfer, and is a set of rules and guidelines about how you should build a web API. (source: https://www.redhat.com/en/topics/api/what-is-a-rest-api)

[11] https://jwt.io

[12] https://swagger.io/docs/specification/v3_0/authentication/bearer-authentication

Additionally, the third section includes an endpoint that allows the updating of data obtained from the PEDs, which the service uses for KPI calculations.

6.1 Energy endpoints specification

The tables throughout this section of the document provide the specifications for the query parameters required for each endpoint.

- **Energy needs**

`[GET]/api/energy/needs`

Retrieves data related to the energy needs of the specified resources, filtered by the given parameters.

Table 4. Parameters for the endpoint to extract energy needs

Parameter	Explanation
resource_id	Specifies the building(s) or resource(s) from which data should be retrieved. A value of 0 or an empty input indicates that data should be obtained for all available buildings. The parameter must support both single and multiple values.
use	Defines the final energy use category. Several options are available: · For thermal energy uses: "heating", "cooling", or "dhw" (domestic hot water). · For aggregated thermal consumption (when individual uses cannot be distinguished): "thermal". · For electrical end uses: "lighting" and "others". · For total electricity consumption (aggregated): "electricity".
scenario	Indicates the scenario for which KPIs should be retrieved. Use 0 to indicate no specific scenario.
startdate	The starting date for the data request period (format: yyyymmdd).
enddate	The ending date for the data request period (format: yyyymmdd).

- **Energy use**

[GET]/api/energy/use

Retrieves data related to the energy use of the specified resources, filtered by the given parameters.

Table 5. Parameters for the endpoint to extract energy use

Parameter	Explanation
resource_id	Specifies the building(s) or resource(s) from which data should be retrieved. A value of 0 or an empty input indicates that data should be obtained for all available buildings. The parameter must support both single and multiple values.
use	Specifies the final use of the energy provided. This parameter has two possible values: · "thermal" – when the energy supplied is of a thermal nature (e.g. heating, cooling, domestic hot water). · "electricity" – when the energy supplied is electrical.
scenario	Indicates the scenario for which KPIs should be retrieved. Use 0 to indicate no specific scenario.
startdate	The starting date for the data request period (format: yyyymmdd).
enddate	The ending date for the data request period (format: yyyymmdd).

- **Delivered energy**

[GET]/api/energy/delivered

Extracts data related to the delivered energy of the specified resources, filtered by the given parameters.

Table 6. Parameters for the endpoint to extract delivered energy

Parameter	Explanation
resource_id	Specifies the resource for which data should be retrieved. This could refer to buildings, energy generation systems, or other monitored entities. A value of 0 or an empty input indicates that data should be retrieved for all available resources.
carrier	Defines the energy carrier used in the system. The accepted values are: · "gas" – Natural gas · "dh" – District heating · "gasoil" – Heating oil · "elec" – Electricity-based heating (e.g., heat pumps) · "biomass" – Biomass fuels · "electricity" – Specifically used for electrical consumption analyses
scenario	Indicates the scenario for which KPIs should be retrieved. Use 0 to indicate no specific scenario.
startdate	The starting date for the data request period (format: yyyymmdd).
enddate	The ending date for the data request period (format: yyyymmdd).

- **Renewable energy**

[GET]/api/energy/renewable

Extracts data related to the renewable energy of the specified resources, filtered by the given parameters.

Table 7. Parameters for the endpoint to extract renewable energy

Parameter	Explanation
resource_id	Specifies the identifier of the renewable energy source from which data is to be retrieved. This parameter must support both single values and multiple values, allowing for the selection of one or several renewable sources simultaneously.
scenario	Indicates the scenario for which KPIs should be retrieved. Use 0 to indicate no specific scenario.
startdate	The starting date for the data request period (format: yyyymmdd).
enddate	The ending date for the data request period (format: yyyymmdd).

- **Stored energy**

[GET]/api/energy/stored

Extracts data related to the energy storage of the specified resources, filtered by the given parameters.

Table 8. Parameters for the endpoint to extract stored energy

Parameter	Explanation
resource_id	Specifies the identifier of the energy storage source from which data should be retrieved. This parameter must support both individual and multiple values, enabling the selection of one or several storage systems.
scenario	Indicates the scenario for which KPIs should be retrieved. Use 0 to indicate no specific scenario.
startdate	The starting date for the data request period (format: yyyymmdd).
enddate	The ending date for the data request period (format: yyyymmdd).

6.2 Data format

Regarding the response data format of the endpoint calls, all of them will return data in the same JSON format. Below is an example of the JSON response format:

```
{
  "resource_id1": {
       "timestamp": 1742463899,
       "value": 11.53
  },
  "resource_id2": {
       "timestamp": 1742463281,
       "value": 7.94
  }
}
```

7 KPI calculation

The calculation of KPIs represents a fundamental step in transforming raw operational data into meaningful metrics that support decision-making and performance evaluation. This process involves the application of standardized formulas and rules to consistently derived datasets, ensuring comparability across different locations, systems, and timeframes. The KPI engine retrieves cleaned and structured data from the relevant sources—whether related to energy consumption, generation, storage, or environmental inputs—and performs a series of transformations to produce quantitative indicators. These include efficiency ratios, energy balances, renewable contribution levels, carbon emissions, or cost-related indicators. The calculated KPIs are then stored centrally and made accessible through visual dashboards, enabling stakeholders to monitor performance trends, detect anomalies, and support strategic planning.

As introduced before, the calculation is centralized in the KPI calculator module that is in charge of implementing the formulas. This module, as anticipated above, receives data in numpy format, which is transformed from the DataFrame coming from the raw data API. This format was specified in Table 1, while the transformed format is depicted in Table 9. Basically, it removes headers or any contextual information and provides timeseries data with pairs timestamp-value.

Table 9. Transformed format for the data used in the KPI calculator

yyyy-mm-dd hh:MI:ss	value0-1	…	value0-N
yyyy-mm-dd hh:MI:ss	value1-1		value1-N
yyyy-mm-dd hh:MI:ss	value2-1		value2-N

The results of the KPIs are stored in a PostgreSQL database. The database schema has been designed to support the storage, categorization, and retrieval of calculated KPIs, along with their contextual and descriptive information. The structure is modular and relational, enabling efficient access through an API layer. It consists of the tables listed in Table 10.

Table 10. KPIs database tables definition

Table	Description
measure	This table stores the actual KPI values calculated for specific resources over defined time periods. Each entry is associated with a particular resource (e.g., a building), a KPI type, and a start date, forming a composite key. The table enables the recording of multiple measurements, potentially across different scenarios, and supports tracking of changes over time.
kpi	This table defines each KPI entity, including its name, reporting frequency, associated type, and measurement unit. It acts as a reference for identifying what each measurement in the measure table represents. The linkage to type and unit ensures semantic clarity and consistency across data.
kpi_type	This table categorises KPIs into broader types or themes, such as efficiency, consumption, emissions, or cost indicators. It facilitates grouping and filtering of KPIs based on their functional or analytical purpose.
unit	The unit table stores the measurement units associated with each KPI (e.g., kWh, °C, $kgCO_2$), allowing results to be correctly interpreted and visualised. It supports both the unit's name and its symbol, ensuring accurate display in dashboards or reports.

The KPI calculator is developed following a generic schema in order to be applicable to any dataset and/or location by just following the requirements in terms of data imputation. Therefore, to decouple the database and the KPI calculator, as identified in Figure 2, there exists a module in charge of interfacing the database (named database API). This module performs two functions: defining the schema with the initial insertion of data and continuous population of the calculated KPIs.

First one of the functions of the KPI database module is the creation of the aforementioned tables and loading the initial parameters. These are the KPI types, units and list of KPIs. Tables 11-13 include the preloaded data at time of writing this deliverable. KPI types and units cover a full list of possible and available values, while the KPIs are only covering energy and energy flexibility dimensions. These will be increased in the life span of the project with other dimensions, when applicable.

Table 11. KPI types loaded into the KPI database

KPI types
energy
energy flexibility
comfort
ICT
environment
mobility
social
economic
citizen engagement

Table 12. KPI units loaded into the KPI database

Name	Symbol
Energy per m2	kWh/m2
Greenhouse emmisions per m2	tCO2/m2
Power	kW
Percentage	%
Energy	kWh
Temperature	ºC
Time (in seconds)	s
Data size	GB
Time (in hours)	hours
Number	#
Number by type	#/type
Number by class	#/class
tons of CO2 equivalent	tCO2eq
kg of CO2 equivalent	kgCO2eq
Speed	Km/h
Time (minute)	min
Energy balance	kWhin/kWhout
No unit	-
dimensionless	dimensionless
Time (in years)	year
Money/Costs	€
Score (1 to 5)	likert
Square meters per year	m2/year
MegaJoules	MJ
Kilogram	kg
Particles	ppm

Table 13. KPIs loaded into the KPI database

KPI type	KPI name	Units
energy	Energy demand / needs thermal	kWh/m2
energy	Energy demand / needs thermal heating	kWh/m2
energy	Energy demand / needs thermal cooling	kWh/m2
energy	Energy demand / needs thermal DHW	kWh/m2
energy	Energy demand / needs electrical	kWh/m2
energy	Energy demand / needs electrical lighting	kWh/m2
energy	Energy demand / needs electrical others	kWh/m2
energy	Energy use thermal	kWh/m2
energy	Energy use electrical	kWh/m2
energy	Delivered / imported energy thermal	kWh/m2
energy	Delivered / imported energy thermal dh carrier	kWh/m2
energy	Delivered / imported energy thermal gas carrier	kWh/m2
energy	Delivered / imported energy thermal gasoil carrier	kWh/m2
energy	Delivered / imported energy thermal electrical carrier	kWh/m2
energy	Delivered / imported energy thermal biomass carrier	kWh/m2
energy	Delivered / imported energy electrical	kWh/m2
energy	Primary energy thermal	kWh/m2
energy	Primary energy thermal dh carrier	kWh/m2
energy	Primary energy thermal gas carrier	kWh/m2
energy	Primary energy thermal gasoil carrier	kWh/m2
energy	Primary energy thermal electrical carrier	kWh/m2
energy	Primary energy thermal biomass carrier	kWh/m2
energy	Primary energy electrical	kWh/m2
energy	CO2 emissions thermal	tCO2/m2
energy	CO2 emissions thermal dh carrier	tCO2/m2
energy	CO2 emissions thermal gas carrier	tCO2/m2
energy	CO2 emissions thermal gasoil carrier	tCO2/m2
energy	CO2 emissions thermal electrical carrier	tCO2/m2
energy	CO2 emissions thermal biomass carrier	tCO2/m2
energy	CO2 equivalent electrical	tCO2/m2
energy	Renewable electricity production	kWh
energy	Renewable thermal energy production solar	kWh
energy	Renewable thermal energy production heat pump	kWh
energy	Surplus thermal energy (exported energy)	kWh

energy	Surplus electrical energy (exported energy)	kWh
energy	Energy balance	kWhin/kWhout
energy flexibility	Stored electrical energy	kWh
energy flexibility	Stored thermal energy	kWh
energy flexibility	Ratio demand vs storage	kWhin/kWhout
energy flexibility	Peak load of electricity demand	kW
energy flexibility	Load profile of electricity demand	kW
energy flexibility	Peak load of thermal (heating/cooling) energy demand	kW
energy flexibility	Load profile of thermal (heating/cooling) energy demand	kW

The next step is to populate the measures table with the calculated KPIs. As stated before, the goal is to provide generic ways to calculate the KPIs, without providing the calculator with logical decisions, while the data filtering is performed in the previous steps of the architecture. It should be noted the KPI calculator is adapted from a previous European project (SmartEnCity[13]). Hence, the naming of the methods differs from ATELIER. Table 14 explains the methods and the description of the KPIs calculated over each one to make it clearer.

Table 14. KPI calculator available functions

Method	Description	Inputs	Outputs
calculateE2E2thE4E6	Method that calculates the indicators computed as the sum of all values found in all columns except the first one (timestamp). For instance, KPIs like total "Energy demand / needs thermal", which is the sum of the carrier-driven KPIs or any of the other carrier-driven KPIs at district level as the sum of the resources / buildings.	Numpy array (Table 7)	Pair date-KPI value
calculateE2carrier	Method that calculates the indicators computed as a normalised value (according to the divisor parameter as input) per energy carrier. Some examples are "Delivered / imported energy thermal gas carrier" or "Energy use thermal" (where the carrier in this case is thermal energy).	Numpy array (Table 7) Divisor	Pair date-KPI value
calculateE4carrierE5E6el	Method that calculates the indicators related to the primary energy and associated CO2 emissions. The multiplier refers to the conversion factors to be used in each specific case.	Numpy array (Table 7) Multiplier	Pair date-KPI value
calculateHourlyMax	Method that calculates the maximum of all received values on an hourly basis (maximum per hour).	Numpy array (Table 7)	Pair date-max value
calculateHourlyMin	Method that calculates the minimum of all received values on an hourly basis (minimum per hour).	Numpy array (Table 7)	Pair date-min value
calculateHourlyAvg	Method that calculates the average of all received values on an hourly basis (average per hour).	Numpy array (Table 7)	Pair date-avg value
calculateRenProd	Method that calculates the renewable contribution from the renewable resources (e.g., photovoltaics).	Numpy array (Table 7)	Pair date-KPI value
calculateBalance	Method that calculates the energy balance by the subtraction of the primary energy KPI calculated before and the renewable production to obtain the positiveness of the building or district.	Numpy array (Table 7)	Pair date-KPI value

[13] https://cordis.europa.eu/project/id/691883/results/es

8 Website-Visualization (PowerBI)

Following the architecture defined before, the last step is the visualization of the KPIs that are stored into the database. For that purpose, the selected tool is PowerBI[14]. Power BI is a data visualization and business intelligence tool developed by Microsoft. It allows users to create interactive dashboards and reports, helping anyone analyze data, monitor performance, and make informed decisions in real time.

Some of the advantages that are provided by the tool are listed below, which justify the use of this technology:

- Interactive Visualization: Offers dynamic dashboards and reports with filtering, drill-down, and cross-highlighting capabilities for intuitive data exploration.
- Real-Time Data Access: Enables connection to live data sources, allowing users to monitor key metrics and trends as they happen.
- Customization and Flexibility: Supports custom visuals, calculated measures (DAX), and personalized views tailored to specific user needs.
- Cloud-Based Sharing: Dashboards and reports can be easily shared via the Power BI Service, with access control and scheduled data update.
- Scalability and Performance: Suitable for both small teams and large organizations, handling large datasets efficiently through features like data compression and in-memory storage.
- Mobile Access: Provides dedicated apps for iOS, Android, and Windows devices, enabling access to insights on the go.
- Pricing: Desktop version is free, being possible to use for any user.

In ATELIER KPIs case, it is employed as the front-end visualization tool, enabling users to explore and interact with KPI data through dynamic dashboards and reports. It connects directly to the centralized KPI database and it generates the cubes (that is a multidimensional data structure that allows for fast analysis of data across multiple dimensions, such as time, location, or category), allowing for real-time visualization of performance metrics, trend analysis, and comparisons across buildings, systems, or energy sources. The platform supports accessibility across desktop and mobile devices, offering custom filtering, drill-down functionality, and automated alerts to support informed decision-making.

Before the preparation of the dashboard, the data cubes should be generated. One example (to be refined in the next months of the project) is drawn in Figure 3. The methodology to do so is as follows:

1. Load data from the database tables.
2. Combine tables to simplify the dimensionality and reduce the 4 aforementioned tables into 2 (KPIs and measures) where data from the types and units are mixed to enrich this metadata. This is recommendable to ensure a star schema[15], according to the PowerBI user's guide.

[14] https://www.microsoft.com/es-es/power-platform/products/power-bi/desktop

[15] https://learn.microsoft.com/en-us/power-bi/guidance/star-schema

3. Create the calendar table, which is recommended by PowerBI good practices for modelers. This table gathers the minimum and maximum dates from all the tables containing this type of data, homogenizes the dates and generates additional useful columns for filtering (e.g., year, quarter, week of year…), which would allow multiple levels of aggregation over all the date type columns.
4. Integrate weather data and building metadata information. Related to weather, existing climate services are integrated to correlate the KPIs with the changing weather conditions over years. In the case of the building metadata, Table 15 summarizes an example for Amsterdam-Republica, with the content that is included in this case. Remarkable to say this information is used both in the calculator and the PowerBI dashboard. In the case of the calculator, it considers per building or resource, the energy carriers and the square meters covered by these carriers, allowing the calculation of the normalized KPIs and the primary energy / CO2 emissions according to each carrier. In terms of visualization, it allows the geolocation of the resources in case of including a map.
5. Finally, following the star schema generation, it should be a fact table (i.e., table with the measurements or, in this case, KPI calculation values), while the rest of tables are dimensions that allow filtering. As well, the relationship between the tables should be created in order to generate the knowledge database (Table 15) so that business intelligence can be applied over the data cube.

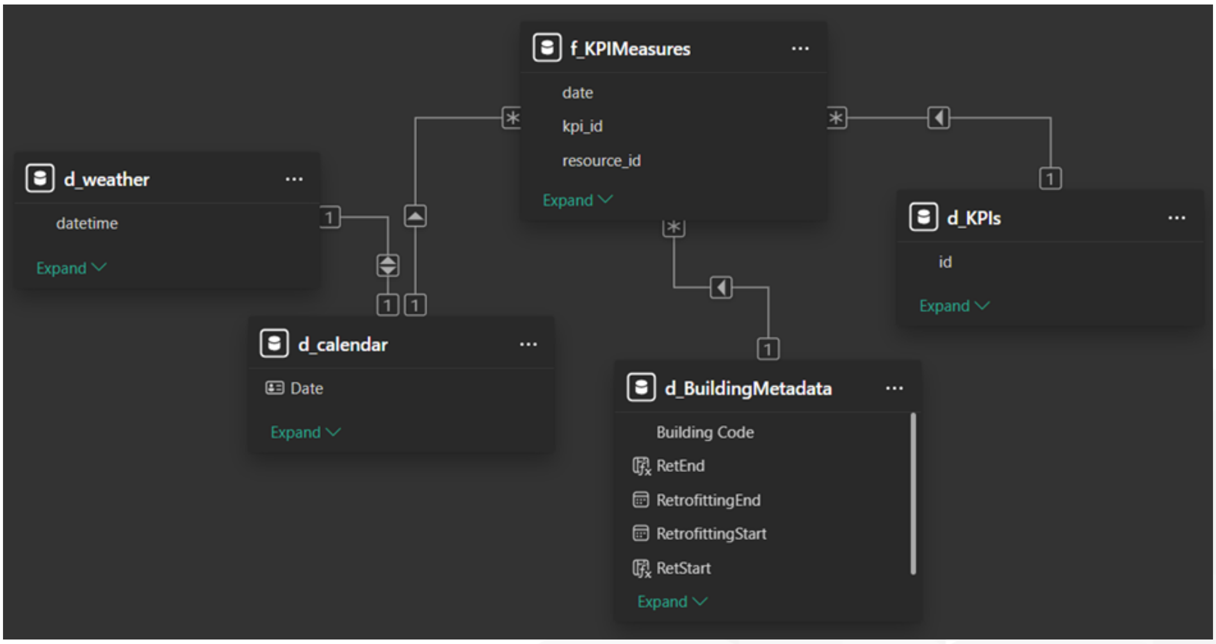

Figure 5. Data cubes generated in PowerBI

Table 15. Building metadata included in the PowerBI data cube

ResourceID	Type	Address / Name	Lat / Lon	carrier1	m2_1	carrier2	m2_2
Building 1	Building	Fat Lady	52.3875, 4.8271	dh	3878	electricity	3878
Building 2	Building	Skinny Lad	52.3870, 4.8269	dh	1957	electricity	1957
Building 3	Building	Short Guy	52.3868, 4.8274	dh	1884	electricity	1884
Building 4	Building	Hotel	52.3872, 4.8278	dh	5765	electricity	5765
Building 5	Building	Small Girl	52.3874, 4.8282	dh	2243	electricity	2243
Building 6	Building	The Joker	52.3866, 4.8270	dh	898	electricity	898
Building 7	Building	Outsider	52.3878, 4.8273	-	-	-	-
PV1	Energy system	Fat Lady	52.3871, 4.8276	electricity	16625	-	-
PV2	Energy system	Skinny Lad	52.3869, 4.8267	electricity	16625	-	-
PV3	Energy system	Short Guy	52.3873, 4.8280	electricity	16625	-	-
PV4	Energy system	Hotel	52.3876, 4.8268	electricity	16625	-	-
PV5	Energy system	Small Girl	52.3867, 4.8275	electricity	16625	-	-
PV6	Energy system	The Joker	52.3870, 4.8281	electricity	16625	-	-
PV7	Energy system	Outsider	52.3879, 4.8272	electricity	16625	-	-

Once the data cube is generated, it is time to draw the analytics in a visual way. The preliminary dashboard is printed in Figure 4. Within this dashboard, 4 main graphs are included:

- Heat consumption where the thermal energy for heating and DHW purposes are identified. These contributions are established in the example per quarter of year, but, thanks to the calendar table, multiple temporal levels are available (from day to year).
- Electricity uses, where it can be identified the net electricity, lighting or any other use, depending on the level of disaggregation of the data; therefore, the KPI calculation. Overlaid, the total renewable generation by electricity sources (e.g., PV) are drawn to allow a visual comparison of the balance.
- Next, a matrix allowing multiple temporal levels (now, in the example, year and quarter), allowing the comparison of the multiple scenarios.
- Last object is a district level analytic providing the global energy use in contrast to the renewable contribution. In the current example, it is observed that 73.98 kWh are coming from renewables from the 147.96 kWh (note these numbers are based on one scenario and one single building of Amsterdam-Republica, without being able to extract conclusions).

At time of writing this delivery, while the database architecture is developed and established, monitoring data is still being collected/post-processed. We demonstrate the database's capability to handle data and perform KPI calculations. The figure illustrates the visulalisation of KPIs based on currently available data. That is to say, data in the explained graphs are not relevant, but it shows the design of the visualization, while changes could occur when data become available and further KPIs are calculated. These modifications will be focused on improving visual information, as well as giving more insights on the KPIs; hence, expanding the current dashboard or replacing the visual objects by more intuitive ones. Moreover, only the energy dashboard is created as data availability is related to this dimension. The rest of dashboards will be created as soon as datasets are collected from the cities.

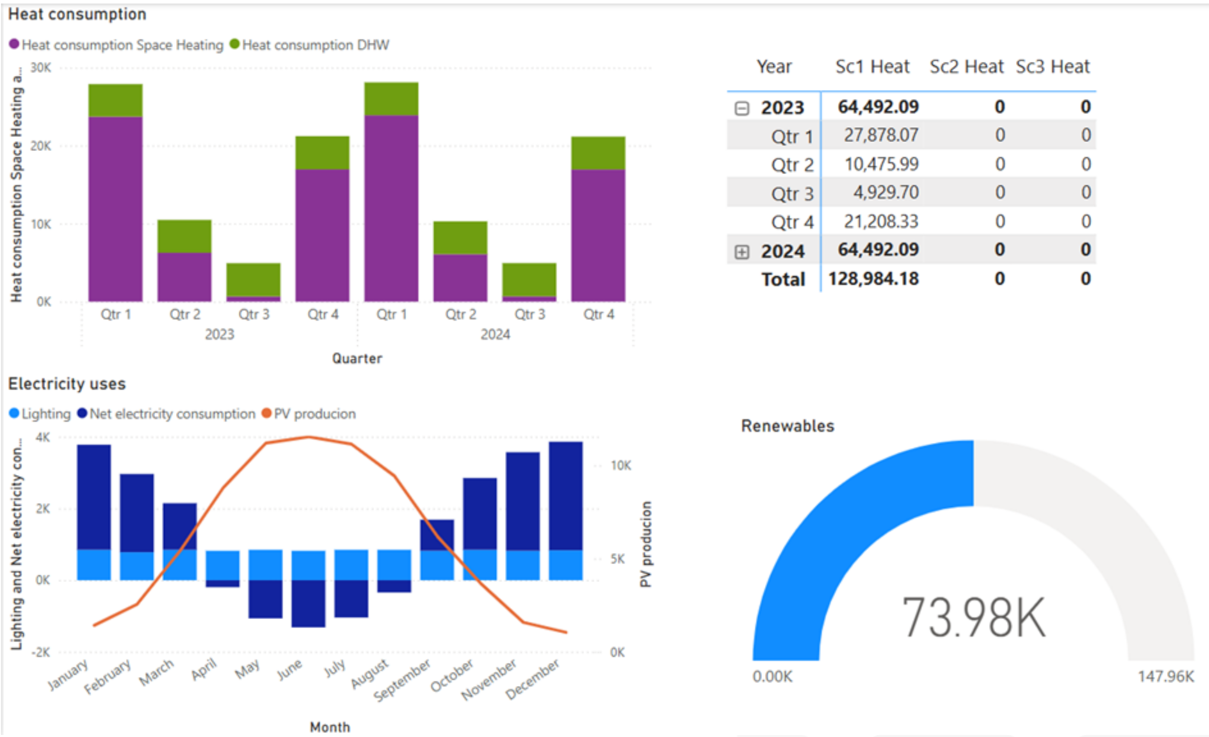

Figure 6. KPI preliminary PowerBI dashboard for KPIs visualization

One of the key advantages of PowerBI is the capability of generating an open and public access to the dashboards. The generated elements allow their integration in a Web page[16]. The design is depicted in Figure 5 for Amsterdam. The landing page provides a short description of the project, the city and a picture. On the left side, the menu facilitates moving through the KPIs dimensions (KPI types) to showcase the dashboards.

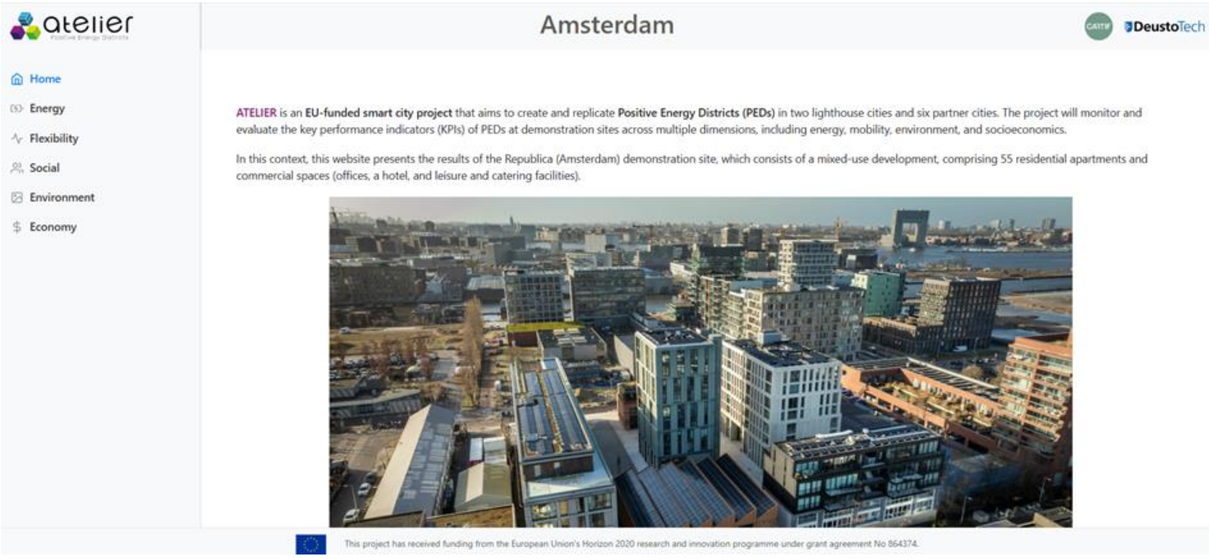

Figure 7. Embedded Web-based visualisation tool for the Amsterdam KPIs

[16] https://amsterdam-kpis.atelier.apps.deustotech.eu/#

9 Website & URL

As mentioned earlier, independent web visualization and API instances are provided for each PED. For the current version of the document, the Amsterdam links are included.

KPIs website: https://amsterdam-kpis.atelier.apps.deustotech.eu

REST API documentation: https://amsterdam-kpis.atelier.apps.deustotech.eu/docs

The deployment and the links related to Bilbao PED will also be available in future versions of this document.

10 Update process

Considering the data flows and data consolidation processes reported in sections 4 and 5, data is updated on a monthly basis, expectedly 3 weeks after the end of each month[17]. Then the data is integrated into the database available through API.

After this process is done, KPI calculations and PowerBI dashboards are updated automatically.

[17] This timing may vary due to holidays and unavailability of researchers/technicians along the data chain.