



Horizon Europe (HORIZON) - HORIZON-CL5-2024-D3-01-11

AI Testing and Experimentation Facility (TEF) for the energy sector – bringing technology to the market

Project no. 101172705



**Large-Scale Testing and Experimentation Facility (TEF) for Assessing,
Validating, and Enhancing AI-Powered Next-Generation Energy Solutions**

D3.1 - EnergyGuard Foundational Framework and services 1st version

By CARTIF

Athens, December, 24th, 2025

Workpackage: WP3 – EnergyGuard TEF Foundation Framework for Data management, DT integration and HPC Services

Authors: CARTIF, ICCS, LXP, HAL, MS

Status: Final

Date: 24/12/2025

Version: 1.0

Classification: PU - Public

Disclaimer:

The EnergyGuard project is co-funded by the Horizon Europe Programme of the European Union under grant agreement No. 101172705. This document reflects only authors' views. The EC is not liable for any use that may be done of the information contained therein.

EnergyGuard Project Profile

Grant Agreement No.: 101172705

Acronym:	EnergyGuard
Title:	Large-Scale Testing and Experimentation Facility (TEF) for Assessing, Validating, and Enhancing AI-Powered Next-Generation Energy Solutions
URL:	http://www.energy-guard.eu
Start Date:	01/01/2025
Duration:	36 months

Partners

	EREVNITIKO PANEPISTIMIAKO INSTITOUTO SYSTIMATON EPIKOINONION KAI YPOLOGISTON (ICCS), Decision Support Systems Laboratory, DSSLab <u>Co-ordinator</u>	Greece
	CENTRO DE INVESTIGACAO EM ENERGIA REN - STATE GRID SA (RDN)	Portugal
	CENTRO DE INVESTIGACIONES ENERGETICAS MEDIOAMBIENTALES Y TECNOLOGICAS (CIEMAT)	Spain
	COMMISSARIAT A L ENERGIE ATOMIQUE ET AUX ENERGIES ALTERNATIVES (CEA)	France
	FUNDACION CARTIF (CARTIF)	Spain
	RIGA MUNICIPAL AGENCY "RIGA ENERGY AGENCY" (REA)	Latvia
	BLUENERGY REVOLUTION SCRL (BER)	Italy
	METASYMBOSE SAS (MS)	France
	LUXPROVIDE SA (LXP)	Luxembourg
	FUNDINGBOX ACCELERATOR SP ZOO (FBA)	Spain
	FORA FOREST TECHNOLOGIES SLL (FORA)	Spain
	FYLLA ANASTASIA (AF)	Greece
	ENGREEN SRL (ENGREEN)	Italy
	BUVNIECIBAS VALSTS KONTROLES BIROJS (SCCB)	Latvia
	HAL SERVICE SPA (HAL)	Italy

Document History

Version	Date	Author (Partner)	Remarks
0.1	02/09/2025	Víctor I. Serna, Susana Martín, José L. Hernández (CARTIF)	Initial ToC
0.2	22/10/2025	All	Initial contributions to the ToC and distribution of sections
0.3	13/11/2025	Víctor I. Serna, Susana Martín, José L. Hernández (CARTIF), Huy Nam Nguyen (MS)	ETLs, Pentaho technology, tests for data collection and digital twin adaptation; Intelligent Query engine
0.4	02/12/2025	Vangelis Karakolis, Theodosios Pountridis, Sotiris Pelekis, Alexandros Tzortzis (ICCS)	Semantics and ontology
0.5	10/12/2025	Gabriele Balzano, Lorenzo Vandoni (HAL), Anastasia Fylla (AF)	Privacy and security mechanisms
0.6	15/12/2025	Frederic Pinel (LXP), José L. Hernández (CARTIF)	Meluxina HPC and final contributions to the general sections of the document
0.65	19/12/2025	Luca Sità (BER), Marta Lafuente (CEDER)	Review
0.7	19/12/2025	José L. Hernández (CARTIF)	Inclusion of comments from reviewers
1.0	23/12/2025	Spiros Mouzakitis (ICCS)	Final revision of the deliverable

Executive Summary

This document delivers the first complete version of the foundational components of the EnergyGuard platform to support testing environments. This includes the creation of a robust data lake with full data management capabilities (curation, lifecycle management, semantic enrichment, harmonisation), the definition of EnergyGuard's semantic data models and ontology, the implementation of a secure and high-performance intelligent query engine, and the adaptation of existing digital twins for seamless and safe integration within the TEF framework. Additionally, it establishes the high-performance computing layer for AI experimentation and defines the end-to-end security and data-privacy framework that will ensure protected data ingestion, processing, and access across all EnergyGuard services.

Overall, this deliverable provides the essential technical foundation required for interoperable data handling, AI testing pipelines, secure digital twin operation, and trustworthy system integration across all EnergyGuard experimentation scenarios.

Table of Contents

1	Introduction	10
1.1	Purpose and Scope.....	10
1.2	Structure of the document.....	11
2	Data management approach and architecture.....	12
2.1	Data lake architecture	12
3	Data extraction, transformation and loading (ETL)	15
3.1	Data requirements	15
3.2	Data collection interfaces from TEF.....	21
3.3	Extraction, Transformation and Loading (ETLs) mechanisms	22
3.3.1	Implementation of the ETLs in the TEFs	24
3.3.2	Initial version of the database.....	27
3.4	Data quality	28
3.5	Conclusions	31
4	EnergyGuard ontology and semantics and alignment with data spaces and AIoD.....	32
4.1	Purpose and Scope.....	32
4.2	Conceptual Structure of the Ontology	32
4.2.1	Reference Ontologies and Standards.....	33
4.2.2	Conceptual Integration with EnergyGuard Components	35
4.3	Integration with European Data Spaces.....	35
4.4	Integration with AI-on-Demand platform	37
4.5	Conclusions	38
5	Data sharing via intelligent query engine	39
5.1	Issues in Energy Management.....	39
5.1.1	Multiplicity of data sources	39
5.1.2	Complexity of decisions	39
5.1.3	Need for rapid and relevant data interrogation.....	39
5.2	Intelligent Query Engine.....	40
5.2.1	Role of the Intelligent Query Engine.....	40
5.2.2	Investigation of Intelligent Query Engines (IQEs).....	40
5.2.3	The Intelligent Query Engine Trino.....	41
5.3	Work Plan.....	41
5.3.1	Single-Server Mode	41
5.3.2	Implementing IQE in an HPC environment.....	42

5.4	Testing on examples of smart grid	42
5.4.1	Purposes of the tests.....	42
5.4.2	Illustration of tests on an example of a smart grid	42
5.5	Conclusions	45
6	Digital twins configuration and adaptation	47
6.1	Interoperability API definition	47
6.2	Analysis of the DTs statuses	48
6.3	Adaptation and customisation of the TEFs DTs	52
6.4	Conclusions	53
7	High-Performance Computing (HPC)	54
7.1	Mapping requirements to Meluxina resources	55
7.1.1	Requirements on Meluxina resources	55
7.1.2	Overall design elements	55
7.1.3	Meluxina-AI impact	56
7.2	Conclusions	56
8	Data security and privacy mechanisms.....	58
8.1	Security requirements	58
8.2	Design of security components.....	59
8.2.1	User Authentication.....	59
8.2.1.1	<i>Objectives and Requirements.....</i>	<i>59</i>
8.2.1.2	<i>Tool Selection and Justification: Keycloak.....</i>	<i>60</i>
8.2.1.3	<i>Authentication Architecture and OIDC Flow</i>	<i>60</i>
8.2.1.4	<i>Identity Lifecycle Management and Security Policies.....</i>	<i>61</i>
8.2.2	User Authorization	61
8.2.3	Auditing	63
8.2.4	Data Anonymization	64
8.2.4.1	<i>Functional and architectural requirements</i>	<i>64</i>
8.2.4.2	<i>Anonymization techniques</i>	<i>65</i>
8.2.4.3	<i>Tools for data anonymization.....</i>	<i>66</i>
8.2.4.4	<i>Tool selection</i>	<i>66</i>
8.2.4.5	<i>Data sources</i>	<i>68</i>
8.2.4.6	<i>Anonymization process.....</i>	<i>68</i>
8.2.5	Data Encryption	69
8.2.5.1	<i>Functional and architectural requirements</i>	<i>69</i>
8.2.5.2	<i>Encryption techniques</i>	<i>69</i>
8.2.5.3	<i>Adopted techniques</i>	<i>69</i>

8.3 Towards implementation of security components	70
8.3.1 Data Anonymization	70
8.3.2 Data Encryption	71
8.4 Data Protection Validation Framework and Assessment of Platform Functionalities	71
8.5 Conclusions	72
9 Conclusion	73
Annex I: References	75

List of Figures

Figure 1: Data lake conceptual approach	12
Figure 2: Data management components architecture.....	13
Figure 3: Methodology for data inventory	15
Figure 4: FTP and non-FTP communication to extract data from TEFs	21
Figure 5: ETL approach for Pentaho Data Integration.....	22
Figure 6: Execution job for the ETLs related to TEFs 2, 3.1, 3.2 and 3.3.....	24
Figure 7: Execution job for the ETL related to TEF 3.4.....	24
Figure 8: Transformation for the ETL related to TEF 2 & 3.3 CEDER-CIEMAT	25
Figure 9: Output of the ETL for TEF 2 & 3.3 CEDER-CIEMAT	25
Figure 10: Transformation for the ETL related to TEF 3.1 CEA	25
Figure 11: Output of the ETL for TEF 3.1 CEA.....	26
Figure 12: Transformation for the ETL related to TEF 3.2 BER	26
Figure 13: Output of the ETL for TEF 3.2 BER.....	26
Figure 14: Transformation for the ETL related to TEF 3.4 CARTIF	27
Figure 15: Output of the ETL for TEF 3.4 CARTIF.....	27
Figure 16: Initial database schema for timeseries storage.....	28
Figure 17: Data quality checker	30
Figure 18: Data quality corrector based on machine-learning	30
Figure 19: Pilot 2 initial data ontology schema	35
Figure 20: Energy anomaly.....	45
Figure 21: API level definition for the communication with the DTs	48
Figure 22: Workflow for the DTs adaptation and customisation.....	52
Figure 23: HPC Architecture (Meluxina)	54
Figure 24: ETL and Meluxina	56

1 Introduction

1.1 Purpose and Scope

The purpose of Deliverable 3.1 (D3.1) is to establish the initial comprehensive foundation of the EnergyGuard platform, enabling robust, secure and interoperable testing environments across the TEF. This deliverable provides a first, integrated version of all core components and services required to support advanced data-driven experimentation, AI model development, and digital twin-based assessments within the project.

More specifically, D3.1 introduces the EnergyGuard data lake alongside all essential data management mechanisms, including data curation, lifecycle management, semantic enrichment and data harmonisation. Together, these elements ensure consistent, high-quality and trustworthy data flows from diverse sources, enabling reliable ingestion, storage and preparation of both structured and unstructured datasets for AI and analytics pipelines.

The deliverable further defines the EnergyGuard semantic data model and associated ontology, establishing the semantic layer that ensures interoperability between heterogeneous energy-related datasets and external ecosystems such as European Data Spaces and AI-on-Demand services. This semantic framework provides the contextualisation required for unified access, interpretation and integration of experimental data.

D3.1 also introduces the first version of the intelligent query engine, designed to deliver secure, high-performance and distributed access to federated data sources. By incorporating advanced SQL engine capabilities and AI-augmented processing techniques, the query engine supports the efficient execution of complex analytical queries, real-time simulations and digital twin interactions on datasets of significant scale.

In addition, the deliverable covers the adaptation and configuration of existing TEF digital twins to ensure their compatibility with the EnergyGuard testing framework. This work enables seamless integration of digital twins into the platform through standardised APIs and provides the necessary configuration rules for their use across varied experimentation scenarios, including the definition of required inputs, outputs and interaction with AI test pipelines.

To support computationally demanding workflows, D3.1 establishes the EnergyGuard High Performance Computing (HPC) layer. This includes the configuration of HPC resources, container-based job execution, workflow orchestration and monitoring tools, ensuring that all AI-related tasks, such as training, validation and inference, can be carried out efficiently, transparently and at scale.

Finally, the deliverable sets out the end-to-end security and data privacy framework that underpins the EnergyGuard platform. This framework ensures secure data handling across all layers, from ingestion to AI model execution, incorporating fine-grained access control, encryption, anonymisation, identity management, penetration testing and ongoing compliance with European data protection regulations.

Overall, D3.1 provides the essential technological, semantic, computational and security foundations upon which all subsequent EnergyGuard experimentation, validation activities and platform services will be developed. It represents a major step towards a unified, secure and high-performance environment for trustworthy AI evaluation within the energy domain.

1.2 Structure of the document

The document begins with the EnergyGuard data management architecture, followed by detailed sections on data extraction, transformation and loading processes. A full description of the EnergyGuard ontology is provided, including its conceptual structure, reference standards and integration with platform components.

Subsequently, the document explains the intelligent query engine, covering its role, design considerations, work plan and initial testing on smart grid examples. The following section addresses the configuration and adaptation of digital twins within the EnergyGuard framework. The High-Performance Computing layer is then described, including its architecture, requirements and operational principles.

Finally, the document concludes with an extensive section on data security and privacy mechanisms, covering authentication, authorisation, auditing, anonymisation, encryption and the Data Protection Validation Framework and Assessment of Platform Functionalities.

In short, the structure of the document contains the next main items:

- Data management approach and architecture
- Data extraction, transformation and loading
- EnergyGuard ontology and semantics
- Data sharing via intelligent query engine
- Digital twins configuration and adaptation
- High-Performance Computing (HPC)
- Data security and privacy mechanisms

2 Data management approach and architecture

This section presents the overall data management approach adopted in EnergyGuard and describes the architecture designed to support secure, interoperable and high-quality data flows across all TEF components. Figure 1 illustrates the core data management workflow that connects the TEF facilities with the EnergyGuard platform through a secure and privacy-preserving Data Lake. Data originating from the TEF repositories enters the system through tailored ETL processes, which extract, transform and load heterogeneous datasets into two complementary storage layers: a Data Warehouse (DWH) for time-series and dynamic information, and a static data repository for contextual or slowly varying datasets. A synchronisation mechanism ensures consistency between these two data domains.

Once stored, data can be accessed through the Intelligent Querying component, which provides enriched and harmonised data views for downstream services. Data access is exposed externally through a dedicated API, enabling seamless integration with Digital Twins and other EnergyGuard components. Throughout the entire workflow, a transversal Security and Privacy layer, covering authorisation, anonymisation and related mechanisms, ensures that data is handled safely and in full compliance with data protection requirements.

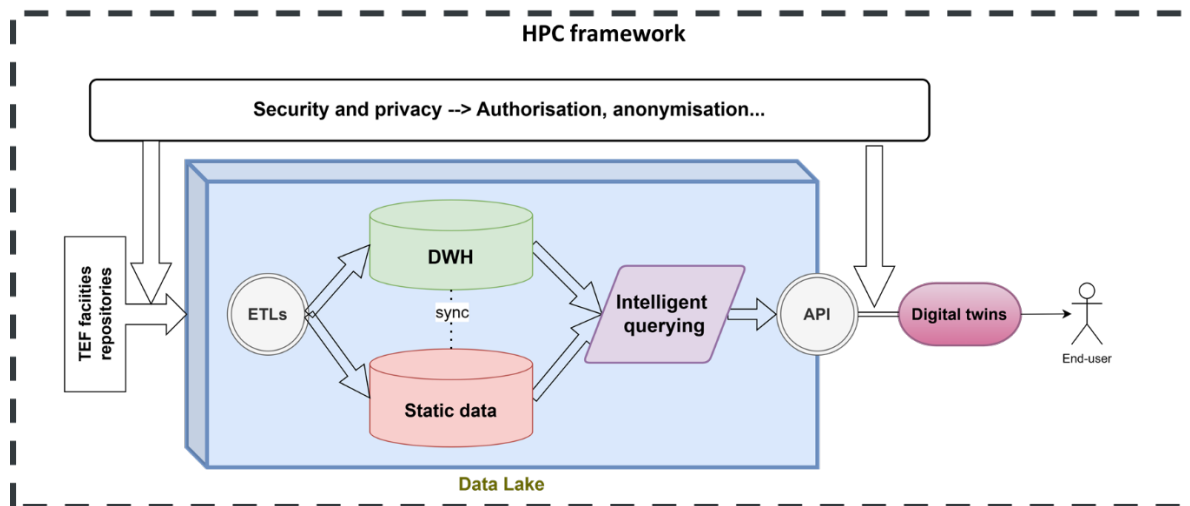


Figure 1: Data lake conceptual approach

2.1 Data lake architecture

Below (Figure 2) is an updated architectural description that summarises the existing description from D2.2 [1], and explicitly incorporates the new components shown in the image but not previously described (i.e., object storage and PostGIS extension). The EnergyGuard data management architecture establishes a modular, interoperable framework designed to manage the complete lifecycle of energy-related data, from its initial ingestion to its secure exchange across federated environments. Each stage of data processing is clearly separated, traceable and aligned with the principles of interoperability, semantics, privacy and security-by-design.

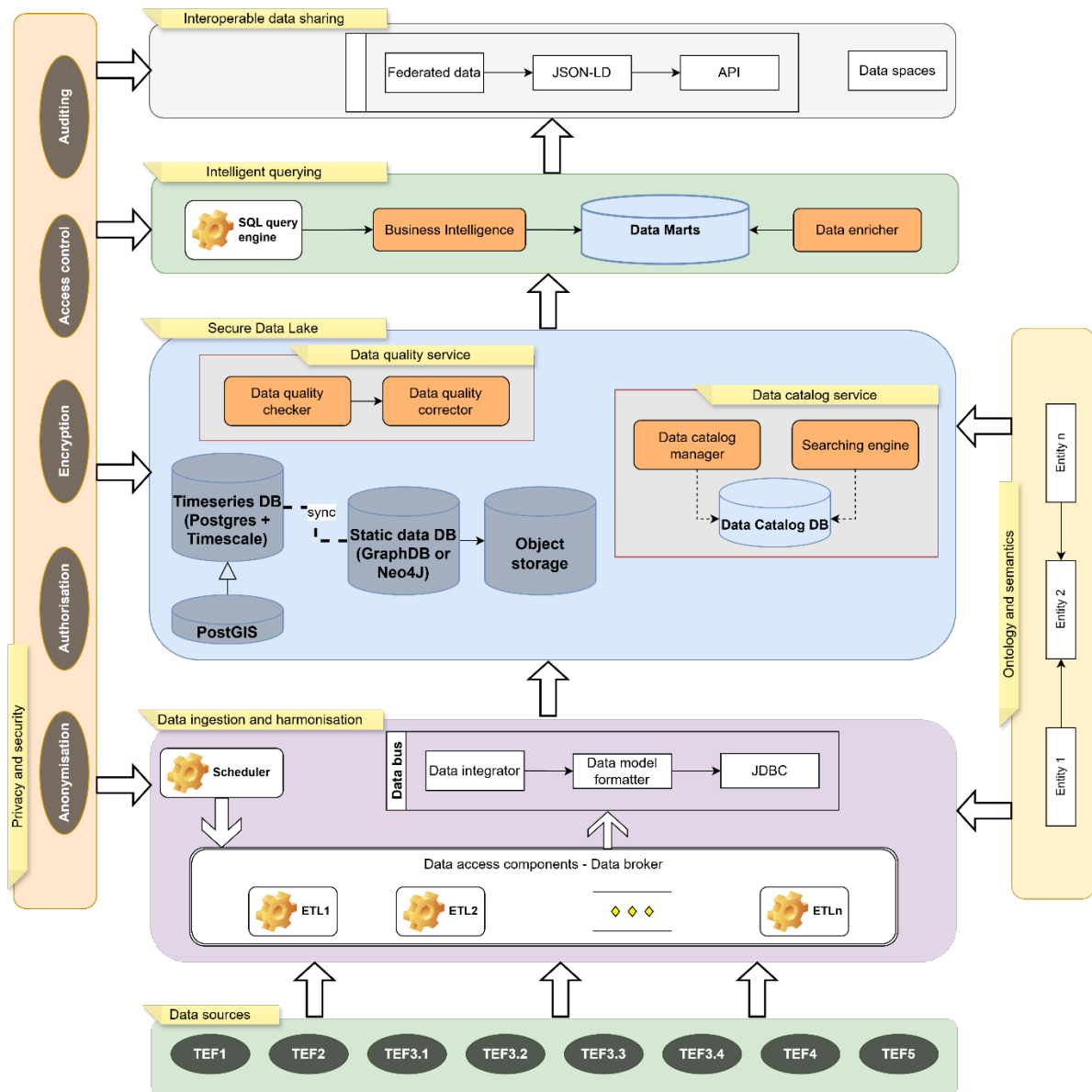


Figure 2: Data management components architecture

Its main layers can be summarised as follows:

1. **Data Sources Layer:** this foundational layer gathers heterogeneous raw data originating from the TEFs. It accommodates multiple protocols, sampling rates and access mechanisms.
2. **Data Ingestion and Harmonisation Layer:** it provides dedicated ETL modules (ETL1...ETLn) triggered by a Scheduler to collect data periodically. These modules collect and harmonise raw data into the EnergyGuard common data model.
3. **Secure Data Lake Layer:** it is responsible for storing, managing and ensuring the quality of time-series and contextual data, but the updated architecture introduces new components not explicitly detailed in D2.2:
 - i. PostGIS extension inside the Timeseries DB to support spatial and geo-referenced data.
 - ii. Object storage, providing scalable handling of unstructured datasets (e.g., files, images, logs, AI inputs).

4. Intelligent Querying Layer: this layer provides semantically enriched access to harmonised data via business intelligence and SQL-based mechanisms.
5. Interoperable Data Sharing Layer: it enables secure, standardised data exchange via APIs, JSON-LD and linked-data technologies.
6. Privacy and Security Transversal Layer: it enforces auditing, access control, authorisation, anonymisation and encryption across all components.
7. Ontology and Semantics Layer: this layer provides semantic alignment and standardised structuring of data through the EnergyGuard ontology.

In summary, the updated architecture preserves the conceptual foundations defined in D2.2, while the new diagram enriches the representation by explicitly incorporating components such as the object storage and PostGIS extension. These additions do not alter the architectural principles but rather strengthen scalability, semantic richness and operational clarity, providing a more complete view of the EnergyGuard data management ecosystem.

3 Data extraction, transformation and loading (ETL)

This chapter outlines the data ingestion, preparation and management processes developed under Task 3.1, which ensure that EnergyGuard has access to high-quality, harmonised and trustworthy data for analytics and AI/ML services. It describes the design of the ETL workflows that collect and process data from diverse local and European ecosystems, the creation of a fully ontology-compliant Data Lake for both structured and unstructured datasets, and the integration of data quality mechanisms to minimise uncertainty.

3.1 Data requirements

First stage in the data extraction, transformation and loading approach is to determine which datasets should be collected. To do so, a data inventory has been made following the methodology depicted in Figure 3. According to the TEF requirements to run services (i.e., AI experiments), energy systems to be monitored and other assets required to run such services, the list of datasets can be obtained.

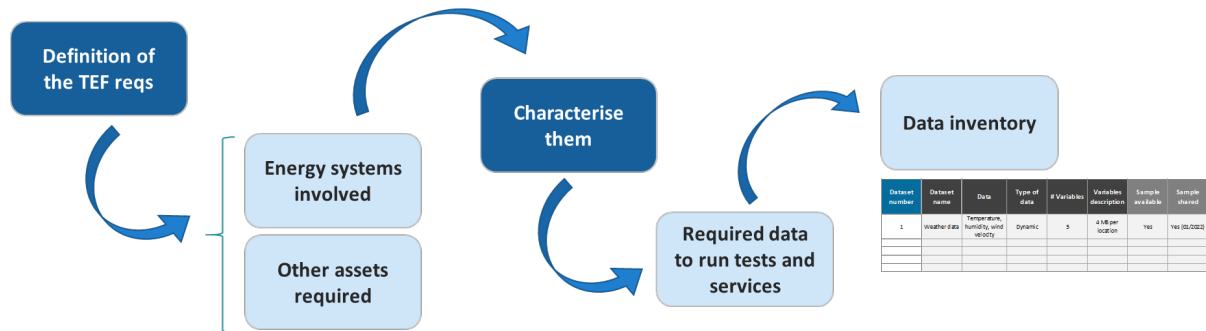


Figure 3: Methodology for data inventory

The results of the methodology are structured in Table 1 that provides a comprehensive inventory of the datasets that will be collected from each TEF involved in the EnergyGuard project. It summarises the key characteristics of the available data, including its origin, structure, acquisition method and intended use within the platform. For each TEF, the table identifies the types of data generated, the associated services that will rely on them, the interfaces and protocols used for data access, and the storage formats and systems in which the data will reside. It also documents operational aspects such as sampling frequency, update mechanisms and expected data volumes, as well as the specific energy systems to which the data relates. This inventory ensures a clear understanding of the data landscape across TEFs and forms the basis for harmonisation, ingestion and subsequent integration into the EnergyGuard data management architecture. In detail, each column of the data requirements extraction process is explained below:

- TEF name: Identifies the Testing and Experimentation Facility (TEF) from which the data is sourced.
- Data to be collected: Describes the specific type of data that will be gathered (e.g., sensor readings, metadata, operational parameters).
- Related Service: Indicates the EnergyGuard service, AI model, or digital twin functionality that will make use of this dataset.
- Interface / Protocol: Specifies the communication method or protocol used to access or retrieve the data (e.g., API, MQTT, Modbus, REST, OPC-UA).

- Data storage system: Defines where the collected data is already stored in the local repositories (e.g., local databases, file storage system, edge storage).
- Format: The structure or encoding of the data (e.g., JSON, CSV, XML, time-series format).
- Data update method: Explains how updates to the data are performed, such as through streaming, batch ingestion, periodic uploads or on-demand extraction.
- Sampling frequency: States how often the data is captured or updated (e.g., every second, every minute, hourly, event-based).
- Estimated size of data: Provides an approximate indication of the volume of data expected over the project duration or per time unit.
- Associated energy system: Identifies the energy-related system or subsystem to which the dataset pertains (e.g., HVAC, PV generation, battery storage, thermal network).

Table 1: Data inventory requirements for the TEFs

TEF name	Data to be collected	Related Service	Interface / Protocol	Data storage system	Format	Data update method	Sampling frequency	Estimated size of data	Associated energy system
1. Digital Twin (DT) of the Portuguese Transmission Network (RDN)	Transmission Grid generation and consumption 15-min data	Digital twin validation	HTTP	Private/ internal webservice	.json	API	Daily	Not yet known, but the Large-Scale Transmission Grid Model has ~1000 buses	Large-Scale Transmission Grid
	Fault simulation data	Fault Analysis	FTP	RDN private storage	.csv	Not applicable	When simulating	~50-100kB	IED such as protective relays
	See CEDER-CIEMAT_Microgrid_Op Data Data to be collected	Model-free voltage estimation	Not known yet	CEDER-CIEMAT_Microgrid_Op Data Output	Not known yet	Not known yet	Not known yet	See CEDER-CIEMAT_Microgrid_Op Data Data Size	CEDER-CIEMAT microgrid
2. CEDER-CIEMAT Microgrid with its Distributed Energy Resources (DERs)	Generation, storage and consumption data from real-time operation of the microgrid	Digital twin validation	Modbus	CIEMAT internal sata server	.csv, .xls	Real-time acquisition with regular batch exports for analysis and external use.	1 second (direct recording); 15 minutes (simple averages without pre-processing)	Approximately 10 MB/day, equivalent to 300 MB/month	CEDER-CIEMAT microgrid, with distributed generation (wind, photovoltaic, mini-hydro), battery storage, charging in buildings and electric vehicle charging stations
	Meteorological conditions (radiation, wind speed, temperature)	Environmental inputs for AI training	Modbus	CIEMAT meteo database	.csv, .xls	Continuous capture by local weather sensors and export at regular intervals.	1 second (direct recording); 15 minutes (simple averages without pre-processing)	Approximately 10 MB/day, equivalent to 300 MB/month	CEDER-CIEMAT weather station integrated in the microgrid system, used for correlation with renewable production and training of predictive models
3.1. Hydrogen testing platforms at CEA LITEN	Operational data from SOC electrolysis system MURPHY, including hydrogen production rate, current density applied, temperature and pressure at different locations...	Data used to validate and tune the Digital Twin of the system	Modbus / CAN / Analog and digital	CEA internal servers	.csv, .txt	Real-time acquisition with an update of data depending on their value variation.	The raw data does not have a sampling frequency but extraction are done with a user specified one and interpolation method.	Depend on the chosen frequency.	CEA Murphy system.

3.2. Hydrogen testing platforms at BER	Operational data from MOSE electrolyzer, including all BOP measurements, system status and faults	AI-based prognostics of electrolyzer	WebHMI / Modbus	BER internal time-series database	.csv, .xls	Continuous acquisition via internal gateway and database sync	up to 0,1s, threshold-based recording	Approx 10MB/day compressed, corresponding to 130MB raw	MOSE electrolyzer
3.3. Hydrogen testing platforms at CEDER-CIEMAT	Operational data from PEM electrolyser integrated in the CEDER-CIEMAT microgrid, including power input, hydrogen production rate, temperature, pressure, system status, and fault logs.	AI-based O&M validation for renewable-powered hydrogen systems	Modbus	CIEMAT internal data logging platform with secure backups and access control	.csv	Continuous acquisition via internal PLCs with periodic batch export.	5-10 seconds, adjusted to PEM system dynamics and sensor availability	5-10 MB/day, which amounts to 150-300 MB/month	PEM electrolyser deployed at CEDER-CIEMAT integrated with renewable generation and storage infrastructure (wind, PV, batteries)
3.4. Hydrogen testing platforms at CARTIF	Meteorological conditions forecasting (radiation, temperature)	Prediction of PV contribution in the AI-based DT for renewable energy optimization	API (external service)	Climate services	JSON	Predicocally collection from the available online climate services	1 hour	185 MB/month	PV system and heat pump for indoor space heating
	Meteorological conditions (radiation, temperature)	Current climate conditions	python RS232 interface	Local CARTIF repositories	csv	Continuous data acquisition	1 min	97 kB/day	PV generation and comfort conditions

	Electrical generation power curve, electrical consumption power curve. Hourly price of electricity. Evolution of stack voltage and current of electrolyser. Evolution of stack voltage and current of the FC. State of the hydrogen storage. Evolution of voltage, current and SOC of the batteries. Evolution of temperatures and pressures in the circuits	H2 hybrid system DT	MQTT	Local CARTIF repositories	csv	Continuous data acquisition	1 min	~300 MB/month	H2 hybrid system operation parameters
4. Riga's multi-apartment residential buildings (GEO RIGA, SCCB CIS)	Energy audits data	for DT validation	TBD	TBD	xls, csv	Automatically generated update	TBD	TBD	(should be precised by ICCS)
	Heat consumption data	for DT validation	TBD	TBD	xls	Monthly added new data	Monthly	TBD	Heating
	List of serie of buildings	for DT validation	TBD	TBD	xls	No update	-	TBD	Heating
	CIS open data	for DT validation	TBD	TBD	csv	Automatically generated update	Daily	TBD	Heating
	Energy efficiency measures	to predict energy efficiency measures and savings	TBD	TBD	csv	No update	-	TBD	Heating
	Meteorological data (historical & normative)	for validation of heat consumption	TBD	TBD	xls or csv	No update	-	TBD	Heating
	LOD data (georiga)	for DT validation	TBD	TBD	gdb, skp, obj, 3ds, gml	No update	-	TBD	Heating

5. Antrodoco Renewable Energy Community (EnGreen)	Consumption data from real-time operation of the REC	Prediction of energy sharing, flexibility services and matching between production and consumption	Meters and More	Engreen private storage	.csv, .xls	API for regular batch exports for analysis and external use.	15 minutes (2G smart meter) or daily (1G smart meter)	TBD	REC Antrodoco consumer members
	Generation, storage and energy fluxes (self-consumption, excess energy, energy shared) data from real-time operation of the REC	Prediction of energy sharing, flexibility services and matching between production and consumption	TBD	Engreen private storage	.csv, .xls	API for regular batch exports for analysis and external use.	15 minutes	TBD	REC Antrodoco prosumer members owner of PV or PV and batteries
	Electric vehicle charging station	Prediction of energy sharing, flexibility services and matching between production and consumption	TBD	Engreen private storage	.csv, .xls	API for regular batch exports for analysis and external use.	15 minutes	TBD	REC Antrodoco members owner of EV charging station

3.2 Data collection interfaces from TEF

Next is to identify the ways to expose data from TEF to collect the previously defined datasets so that ETLs could extract and load data into the data lake. Table 2 summarises the protocols, frequency for accessing data, format and method to gather available data from TEFs. Some of them are still under definition, so, for this first release of the ETLs, the available ones have been developed, while next releases will include the rest of TEFs.

Table 2: Interfaces to collect data from the TEFs

TEF	Protocol	Frequency	Data format	Method
1. Digital Twin (DT) of the Portuguese Transmission Network (RDN)	API (HTTP)	1 day	TBD	TBD
2. CEDER-CIEMAT Microgrid with its Distributed Energy Resources (DERs)	FTP	1 day	.csv	sFTP
3.1. Hydrogen testing platforms at CEA LITEN	FTP	1 day	.csv	sFTP
3.2. Hydrogen testing platforms at BER	SFTP	1 day	Influxdb LP (gzipped)	sFTP
3.3. Hydrogen testing platforms at CEDER-CIEMAT	FTP	1 day	.csv	sFTP
3.4. Hydrogen testing platforms at CARTIF	InfluxDB-API	On demand	JSON	POST
4. Riga's multi-apartment residential buildings (GEO RIGA, SCCB CIS)	pdf, csv, xml, gis files, other	Mixed	Various formats	TBD
5. Antrodoco Renewable Energy Community (EnGreen)	TBD	15 min	TBD	TBD

As conclusion of the table above, there exist two main groups, a) FTP-based communication; b) non-FTP communication. Then, as illustrated in Figure 4, an intermediate FTP server is hosted in CARTIF facilities to push all the csv files. The advantage of this approach is reducing the complexity of the ETLs for connection, because one single connection is able to obtain all the csv files.

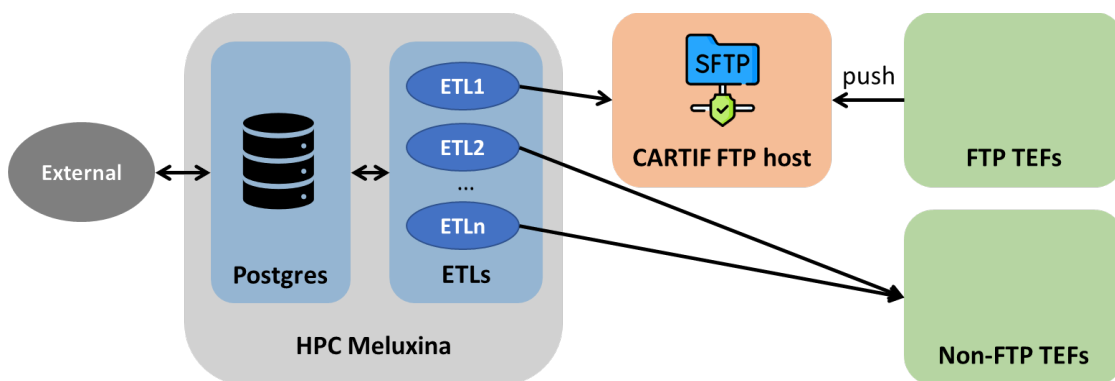


Figure 4: FTP and non-FTP communication to extract data from TEFs

3.3 Extraction, Transformation and Loading (ETLs) mechanisms

For the implementation of the ETL processes, Pentaho Data Integration (PDI) [2] is selected as technology. This is an open-source data integration platform widely used for building scalable and maintainable data pipelines. PDI provides a graphical development environment (Spoon) that enables the design of complex workflows through a rich set of pre-built components for extraction, cleansing, enrichment, transformation and loading. Its flexibility, support for multiple data formats and ability to orchestrate both batch and streaming processes make it well suited for handling the heterogeneous datasets generated by the TEFs.

PDI, also known as Kettle, enables the development of robust ETL workflows through two main artefact types: Transformations and Jobs. Transformations define the actual data-processing logic, while Jobs orchestrate the execution order, dependencies, control flow and scheduling of multiple transformations or external processes. This workflow is depicted in Figure 5 and reflects the sequence of operations applied to incoming raw data before it is consolidated in the Data Lake. Data first enters the ETL pipeline according to the frequency of the job execution. Then, filtering mechanisms remove irrelevant or invalid records. Subsequent transformation steps reshape, convert and harmonise the data according to the EnergyGuard common data model, including operations such as type mapping, unit conversion, aggregation or structural alignment. Additional logic components manage branching, conditional flows and the orchestration of multiple processing paths. Once transformed, the cleaned and harmonised dataset is structured into a consistent tabular representation, which is finally loaded into the Data Lake for secure storage and further use by digital twins, AI pipelines and other EnergyGuard services.

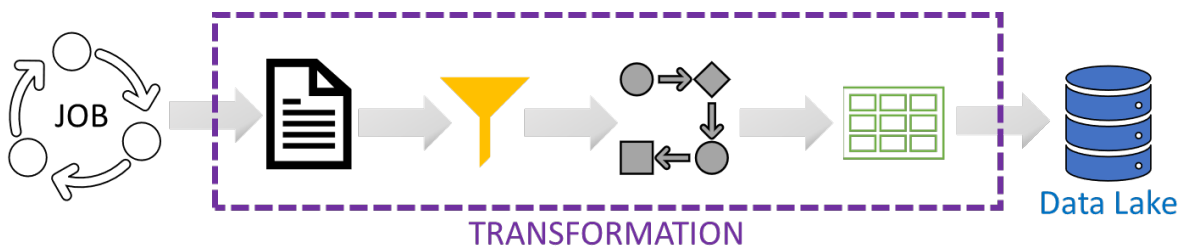


Figure 5: ETL approach for Pentaho Data Integration

Within a typical EnergyGuard ETL pipeline, the following technical steps are involved:

1. Extraction: PDI provides numerous input components capable of ingesting data from varied TEF sources, including:
 - a. File-based inputs (CSV, JSON, XML, Excel, Parquet).
 - b. Database connectors (PostgreSQL, MySQL, SQL Server, Oracle, GraphDB, Neo4J through JDBC).
 - c. REST and web services (via the REST Client or HTTP Client steps).
 - d. Streaming inputs (MQTT, Kafka).
 - e. Custom scripts (JavaScript, Python) for non-standard sources.

These extraction steps preserve metadata and allow schema inference or explicit mapping where required.

2. Pre-processing and Filtering: Once extracted, raw data passes through pre-processing steps designed to ensure structural and syntactic integrity. In PDI, this commonly includes:

- a. Filter Rows to exclude invalid or irrelevant records.
- b. Select Values to rename, remove, or convert fields.
- c. Replace in String or String Operations for text normalisation.
- d. Number Range, Regex Evaluation, or Validator to enforce domain constraints.
- e. Null Handling (e.g., If Null, Coalesce).

These steps prepare the dataset for semantic harmonisation and reduce inconsistencies before transformation.

3. Transformation and Harmonisation: This stage adapts the cleaned data to the EnergyGuard common data model. PDI provides powerful functions for structural transformations, such as:
 - a. Mapping and lookup operations using Database Lookup, Stream Lookup, or Stream Join.
 - b. Type conversion (Date to timestamp, numeric scaling, boolean mapping).
 - c. Unit conversions implemented via scripting or user-defined functions.
 - d. Aggregation and summarisation using Group By, Memory Group By or Aggregator.
 - e. Pivoting or unpivoting with Row Normaliser or Denormaliser steps.
 - f. Branching and conditional flows via Switch/Case, Filter Rows, or Modified JavaScript Value.

These operations ensure that heterogeneous datasets from different TEFs align with a coherent and interoperable schema.

4. Data Quality and Enrichment: To guarantee trustworthy input for AI/ML workflows, additional steps ensure data completeness, accuracy and semantic value:
 - a. Data validation using Data Validator.
 - b. Data imputation via built-in functions (mean/median fill) or Python scripts.
 - c. Metadata enrichment (adding tags, units, device identifiers).
 - d. Temporal alignment using Windowing, Sort Rows, or Time Series utilities.
 - e. Integration with ontological metadata, referencing lookup tables or semantic mappings.

These steps directly contribute to the EnergyGuard Data Quality Service and support reliable downstream analytics.

5. Structuring and Formatting: Once harmonised, the dataset is transformed into a consistent, tabular representation suitable for loading:
 - a. Row to JSON or JSON Output for NoSQL or API-based storage.
 - b. Table Output, Bulk Loader, or Insert/Update for relational databases.
 - c. Graph database writers (custom JDBC steps for Neo4J or GraphDB).
 - d. Metadata injection when building dynamic transformations for multiple TEFs.

This stage ensures compatibility with the Data Lake storage technologies.

6. Loading into the Data Lake: Finally, the processed dataset is written into the appropriate storage tier:
 - a. TimescaleDB/PostgreSQL for time-series data using Table Output or Bulk Loader.
 - b. Static repositories (GraphDB/Neo4J) using JDBC or REST endpoints.
 - c. Object storage (S3, MinIO, or local file system) using file output steps.
 - d. Batch orchestration through Jobs that coordinate multiple loading tasks.

3.3.1 Implementation of the ETLs in the TEFs

As specified in Table 2, seven ETLs should be developed (considering the TEF 2 and TEF 3.3, related to CEDER-CIEMAT, provide a single dataset for both cases). However, the cases of TEF 1 (Digital Twin (DT) of the Portuguese Transmission Network) and TEF 4 (Riga's multi-apartment residential buildings) and TEF 5 (Antrodoco Renewable Energy Community) are still under definition. Therefore, within this first release, the ETLs related to the TEFs 2 (CEDER-CIEMAT Microgrid with its Distributed Energy Resources (DERs)), 3.1 (Hydrogen testing platforms at CEA LITEN), 3.2 (Hydrogen testing platforms at BER), 3.3 (Hydrogen testing platforms at CEDER-CIEMAT) and 3.4 (Hydrogen testing platforms at CARTIF) are developed. From all of the already defined interfaces, all of them with the exception of TEF 3.4 are providing data via FTP connection; hence, one single job is enough.

Starting with this case, Figure 6 illustrates the single job to download files from the FTP server. The "Start" step executes once per day and; then, connects each subfolder of the FTP folder to download the files. In the case of the TEF 3.2 BER, files are zipped; therefore, an "Unzip file" step is required previously. Having the files locally downloaded, each transformation is executed. It should be stated the transformation cannot be unique as each csv file has its own structure; hence, transformation should be adapted in each individual case. The last step is moving the files to a backup local folder to keep traceability of the datasets. The case of the TEF 3.4 CARTIF (see Figure 7) is much simpler as it simply starts the process, set the variable (connection parameters to the InfluxDB-API) and; then, executes the transformation of data.

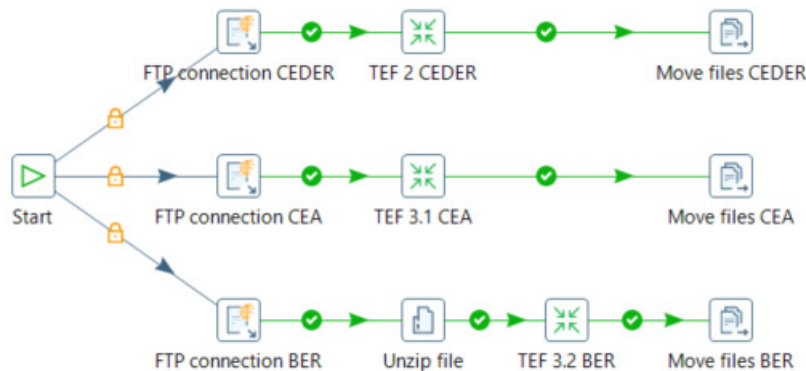


Figure 6: Execution job for the ETLs related to TEFs 2, 3.1, 3.2 and 3.3



Figure 7: Execution job for the ETL related to TEF 3.4

As stated before, the job is executed periodically, according to a pre-set frequency, depending on the sampling and update frequencies of the TEFs. The first case, TEF 2 & 3.3 CEDER-CIEMAT, provides data once per day via FTP. Once files are downloaded; then, the corresponding transformation is launched, which is detailed in Figure 8. This transformation is very simple as it only requires filtering, splitting variables and generating the final output that is illustrated in Figure 9. As observed, it provides three columns: timestamp, sensor_id and value, which will be mapped into the final database structure, following a timeseries classification.



Figure 8: Transformation for the ETL related to TEF 2 & 3.3 CEDER-CIEMAT

timestamp	sensor_id	f_value
2025-10-19 00:00:00	ACTARIS	46220
2025-10-19 00:00:00	ARFRISOL	12503
2025-10-19 00:00:00	ATLANTIC	32
2025-10-19 00:00:00	BATERIA_LITIO	0
2025-10-19 00:00:00	BATERIA_NMC_PEPa_II	0
2025-10-19 00:00:00	BATERIA_PEPa_I	0
2025-10-19 00:00:00	BATERIA_PEPa_II	0
2025-10-19 00:00:00	BATERIA_REDOX	0
2025-10-19 00:00:00	BOMBAS	0
2025-10-19 00:00:00	CENTRAL_TERMICA	0
2025-10-19 00:00:00	DEPOSITO_NIVEL	1610
2025-10-19 00:00:00	DEPOSITO_NIVEL_INFERIOR	1170
2025-10-19 00:00:00	DEPOSITO_NIVEL_LEVI	490
2025-10-19 00:00:00	ED02	0
2025-10-19 00:00:00	ED02_1	8263
2025-10-19 00:00:00	ED03	0
2025-10-19 00:00:00	ED03_1	14247

Figure 9: Output of the ETL for TEF 2 & 3.3 CEDER-CIEMAT

Next TEF is 3.1 CEA, whose transformation and output are depicted in Figure 10 and Figure 11, respectively. The case of the FTP files uploaded in this TEF requires the generation of the date because the data format only includes the seconds between consecutive samples. Once full timestamp is created, non-useful rows are removed in order to transform the dataset into the same format that was explained in the TEF before (i.e., timeseries structure).

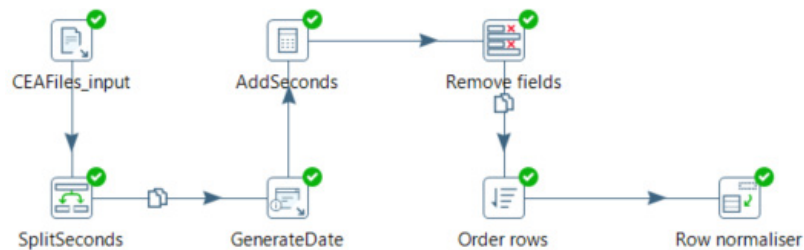
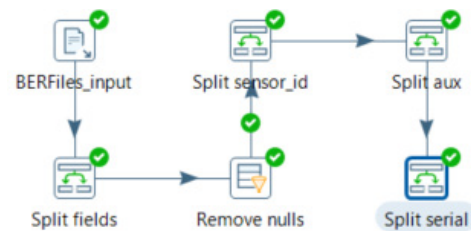


Figure 10: Transformation for the ETL related to TEF 3.1 CEA

finalTime	sensor_id	f_value
2025/11/21 10:43:19.761000000	TT363	716.2
2025/11/21 10:43:19.761000000	TT303	658.6
2025/11/21 10:43:19.761000000	TT320	658.6
2025/11/21 10:43:19.761000000	TT304	700.1
2025/11/21 10:43:19.761000000	PT306	39.5
2025/11/21 10:43:19.761000000	PT353	6.4
2025/11/21 10:43:19.761000000	TT402	20
2025/11/21 10:43:19.761000000	TT404	21.9
2025/11/21 10:43:19.761000000	TT406	23.4
2025/11/21 10:43:19.761000000	TT408	232
2025/11/21 10:43:19.761000000	TT410	383
2025/11/21 10:43:19.761000000	TT414	20
2025/11/21 10:43:19.761000000	TT415	330
2025/11/21 10:43:19.761000000	TT419	23.4
2025/11/21 10:43:19.761000000	TT422	649.4
2025/11/21 10:43:19.761000000	DPT401	0.7
2025/11/21 10:43:19.761000000	DPT403	0.2
2025/11/21 10:43:19.761000000	DPT411	0
2025/11/21 10:43:19.761000000	PT407	39.9
2025/11/21 10:43:19.761000000	PT418	6.3
2025/11/21 10:43:19.761000000	PT420	1013.2

Figure 11: Output of the ETL for TEF 3.1 CEA

The case of BER (Figure 12 and Figure 13) is a very simple transformation, where basically splitting fields and removing null values (due to errors in the monitoring file) obtain the result in the specific format of the database. Missing is the connection to the database, which is not deployed yet and it will be part of the next release of the project.

**Figure 12: Transformation for the ETL related to TEF 3.2 BER**

aux	serial	serial_number	sensor_id	f_value	timestamp
aux	serialNumber	1	FAN_T103	0	1733443200900000000
aux	serialNumber	1	FAN_T103	0	1733443231000000000
aux	serialNumber	1	FAN_T103	0	1733443261100000000
aux	serialNumber	1	FAN_T103	0	1733443291200000000
aux	serialNumber	1	FAN_T103	0	1733443321300000000
aux	serialNumber	1	FAN_T103	0	1733443351500000000
aux	serialNumber	1	FAN_T103	0	1733443381400000000
aux	serialNumber	1	FAN_T103	0	1733443411500000000
aux	serialNumber	1	FAN_T103	0	1733443441600000000
aux	serialNumber	1	FAN_T103	0	1733443471600000000
aux	serialNumber	1	FAN_T103	0	1733443501700000000
aux	serialNumber	1	FAN_T103	0	1733443531700000000
aux	serialNumber	1	FAN_T103	0	1733443561800000000
aux	serialNumber	1	FAN_T103	0	1733443591800000000
aux	serialNumber	1	FAN_T103	0	1733443621800000000
aux	serialNumber	1	FAN_T103	0	1733443651900000000
aux	serialNumber	1	FAN_T103	0	1733443681900000000
aux	serialNumber	1	FAN_T103	0	1733443712000000000
aux	serialNumber	1	FAN_T103	0	1733443742000000000
aux	serialNumber	1	FAN_T103	0	1733443772000000000
aux	serialNumber	1	FAN_T103	0	1733443802100000000

Figure 13: Output of the ETL for TEF 3.2 BER

Last but not least, in this release, the TEF 3.4 CARTIF ETL is developed, which differs from the previous ones as it is using the Influx-DB API to connect instead of FTP. Then, the transformation should generate the HTTP requests. On the one hand, to obtain the list of available variables / measurements. On the other hand, to obtain the real values for each variable. With the result of the HTTP query, the non-desirable fields are deleted (i.e., mainly related to the query parameters), while final formatting is prepared to adopt the timeseries one.

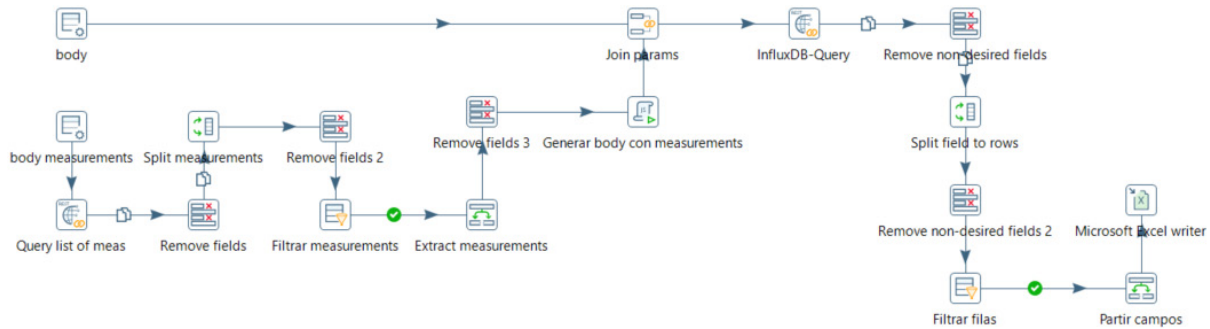


Figure 14: Transformation for the ETL related to TEF 3.4 CARTIF

#	col0	result_col	table	_time	_value	_field	_measurement	line_nr
1		_result	0	2025-11-27T08:17:36.131Z	0	fieldValue	ALTA_PRESION_MH1_Alarm	2
7..		_result	0	2025-11-27T08:17:36.151Z	0	fieldValue	ALTA_PRESION_MH2_Alarm	2
2		_result	0	2025-11-27T08:17:41.245Z	0	fieldValue	ALTA_PRESION_MH1_Alarm	3
7..		_result	0	2025-11-27T08:17:41.265Z	0	fieldValue	ALTA_PRESION_MH2_Alarm	3
3		_result	0	2025-11-27T08:17:50.889Z	0	fieldValue	ALTA_PRESION_MH1_Alarm	4
7..		_result	0	2025-11-27T08:17:50.909Z	0	fieldValue	ALTA_PRESION_MH2_Alarm	4
4		_result	0	2025-11-27T08:17:54.992Z	0	fieldValue	ALTA_PRESION_MH1_Alarm	5
7..		_result	0	2025-11-27T08:17:55.012Z	0	fieldValue	ALTA_PRESION_MH2_Alarm	5
5		_result	0	2025-11-27T08:17:59.779Z	0	fieldValue	ALTA_PRESION_MH1_Alarm	6
7..		_result	0	2025-11-27T08:17:59.799Z	0	fieldValue	ALTA_PRESION_MH2_Alarm	6
6		_result	0	2025-11-27T08:18:04.853Z	0	fieldValue	ALTA_PRESION_MH1_Alarm	7
7..		_result	0	2025-11-27T08:18:04.873Z	0	fieldValue	ALTA_PRESION_MH2_Alarm	7
7		_result	0	2025-11-27T08:18:09.996Z	0	fieldValue	ALTA_PRESION_MH1_Alarm	8
7..		_result	0	2025-11-27T08:18:10.016Z	0	fieldValue	ALTA_PRESION_MH2_Alarm	8
8		_result	0	2025-11-27T08:18:14.532Z	0	fieldValue	ALTA_PRESION_MH1_Alarm	9
7..		_result	0	2025-11-27T08:18:14.547Z	0	fieldValue	ALTA_PRESION_MH2_Alarm	9
9		_result	0	2025-11-27T08:18:19.705Z	0	fieldValue	ALTA_PRESION_MH1_Alarm	10
7..		_result	0	2025-11-27T08:18:19.725Z	0	fieldValue	ALTA_PRESION_MH2_Alarm	10
1..		_result	0	2025-11-27T08:18:24.641Z	0	fieldValue	ALTA_PRESION_MH1_Alarm	11
7..		_result	0	2025-11-27T08:18:24.661Z	0	fieldValue	ALTA_PRESION_MH2_Alarm	11
1..		_result	0	2025-11-27T08:18:29.662Z	0	fieldValue	ALTA_PRESION_MH1_Alarm	12
7..		_result	0	2025-11-27T08:18:29.682Z	0	fieldValue	ALTA_PRESION_MH2_Alarm	12
1..		_result	0	2025-11-27T08:18:34.718Z	0	fieldValue	ALTA_PRESION_MH1_Alarm	13

Figure 15: Output of the ETL for TEF 3.4 CARTIF

3.3.2 Initial version of the database

The timeseries database has been developed following a star-schema approach recommended in data science practices (e.g., Microsoft PowerBI [5], TowardsDataScience [6]). A star schema is a relational model widely used in data warehousing and business intelligence because it improves query performance and simplifies data organization. Its structure resembles a star: a central fact table connected to multiple dimension tables. The main components are:

- Fact table: It is the core table storing quantitative and measurable data, such as temperature or energy readings. These tables typically contain many rows and use foreign keys to link to dimension tables. Their content is predominantly numerical, while contextual information

(timestamps, text, categories) is placed in dimension tables or represented only through primary keys.

- Dimension tables: Tables providing descriptive context for the facts, for example time, sensor identifiers, or location details. They enrich the fact table by enabling meaningful queries, filtering, and reporting.

According to this, a relational database has been designed for storing timeseries data, as represented in Figure 16. The schema consists of four main tables: one fact table (f_tsData) storing raw numerical data, and three dimension tables (d_*) for calendar dates, sensors, and sensor types. Although timestamps are often stored directly in the fact table, business intelligence best practices recommend separating them into a dedicated date dimension. This improves aggregation, filtering, and consistency across queries.

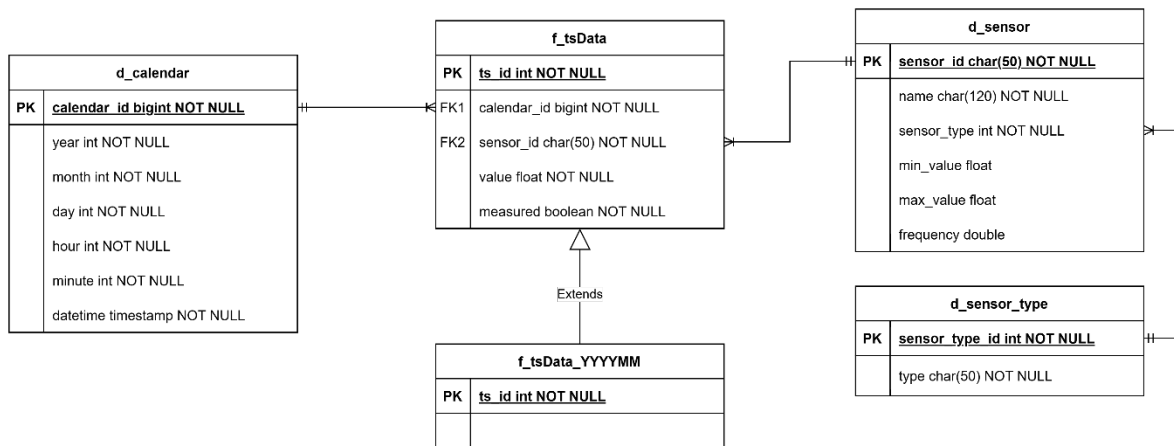


Figure 16: Initial database schema for timeseries storage

The schema also includes an additional table related to an inheritance principle. Monthly tables (e.g., f_tsData_YYYYMM) inherit from the main fact table (f_tsData). This approach mimics the logic of hypertables and chunks in TimescaleDB (a well-known extension of Postgres), but does not require a timestamp column or inclusion of the timestamp in the primary key, reducing complexity according to the business intelligence principle.

3.4 Data quality

Data quality refers to the degree to which a dataset is fit for its intended purpose, reflecting its accuracy, completeness, consistency, timeliness, validity and integrity. High-quality data reliably represents the real-world conditions it describes, is free from errors or inconsistencies, and adheres to defined standards and formats. In the context of analytics, AI/ML pipelines and digital twin simulations, data quality ensures that inputs are trustworthy, interpretable and suitable for generating robust and reproducible results. Effective data quality management involves systematic processes for detecting, correcting and preventing issues that may compromise the usability or reliability of the data.

In the context of EnergyGuard, data quality is assessed through a comprehensive, multi-dimensional framework. Seven key dimensions are considered, each capturing a different aspect of the suitability and reliability of the data. Among these, accuracy, completeness and consistency play the most critical roles. Table 3 provides an overview of the dimensions and the specific approach adopted for each within the project.

Table 3: Data quality approach in the EnergyGuard project

Data quality dimension	EnergyGuard approach
Accuracy	Detection of values outside expected ranges
Completeness	Gap identification and data interpolation
Reliability	Combination of accuracy and completeness
Consistency	Identification of normal behavioural patterns
Relevance	Filtering of non-essential or service-irrelevant data
Accessibility	Use of open APIs and communication protocols
Timeliness	Assessment of data freshness through pre-analytics

Each one of the dimensions can be defined as follows:

- Accuracy assesses the proportion of data points that fall within expected operational ranges, as defined by experts. These expected ranges differ from the sensor's technical limits; values outside them may be valid but are treated as outliers and excluded from service use.
- Completeness measures the amount of missing data by comparing the number of actual readings with the number of samples that should have been collected based on the sampling frequency.
- Reliability provides an overall indication of dataset robustness by combining accuracy and completeness.
- Consistency checks whether data follow typical statistical patterns, such as values clustering around the mean within standard deviation boundaries.
- Relevance determines how much of the collected data is truly useful for the intended services, filtering out non-essential attributes.
- Accessibility evaluates the ability to retrieve and share data, considering the use of open protocols and APIs, as well as any limitations in accessing specific datasets.
- Timeliness reflects how up-to-date the data are and the delay between measurement and availability, ensuring that information is sufficiently fresh for decision-making.

From these seven dimensions, relevance, accessibility and timeliness are not dynamically calculated as they do not depend on the datasets entering into the ETL process, but the available data, access and sampling frequency in the local repositories.

Data quality approach relies on a previous project, where the methodology was developed and applied. A twofold approach was implemented: 1) data checker to determine the quality of a dataset based on four indicators (i.e., completeness, accuracy, reliability and consistency); 2) data corrector where machine-learning techniques allow the imputation of data. Within EnergyGuard, this approach will be improved and enhanced to reduce uncertainties in the calculation and data imputation methods.

The checker [3] operates through two possible pathways, as shown in Figure 17. In the first, raw data files from the pilots (e.g., CSV, Excel, JSON) are ingested by a loading module that normalises formats and transforms them into a unified structure based on a Python DataFrame, guided by a user-defined configuration file. In the second pathway, data are obtained through collection tools such as Pentaho or custom Python/Java scripts, which already handle loading and filtering, sending the prepared data directly to the statistics module. In both cases, the statistics module computes the full set of data quality indicators across all defined dimensions using the standardised DataFrame format, ensuring interoperability, reproducibility and scalability regardless of the data source.

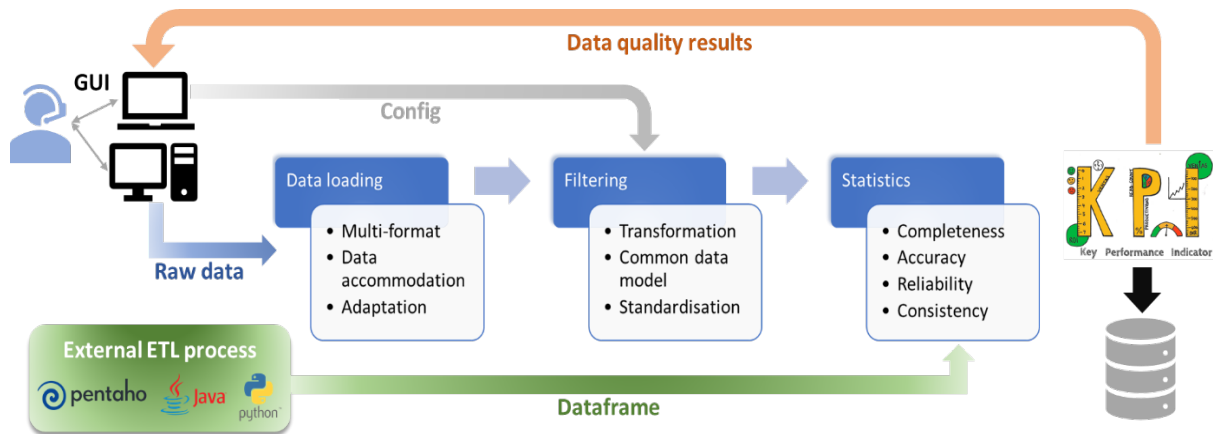


Figure 17: Data quality checker

The data corrector [4] employs machine-learning techniques to improve key data quality dimensions, in particular completeness, accuracy and consistency, by reconstructing missing or inconsistent values. After removing duplicates and assessing completeness, the correction mechanism is triggered whenever data availability falls below a 90% threshold. The service operates asynchronously and only activates once enough historical data has been collected to ensure stable and meaningful model performance. Minimum data requirements depend on the operational characteristics of each system (e.g., full seasonal cycles for HVAC systems).

The corrector architecture (Figure 18) includes a configuration module, which specifies the variables to be corrected, and a learning module, which retrieves historical data from the Data Warehouse to train predictive models. Two types of machine-learning models are used: linear regression for cumulative variables such as energy consumption, ensuring consistency in progressive values; and random forest regression for non-cumulative variables like temperature, capturing non-linear relationships effectively. Trained models generate corrected values that populate new datasets, clearly labelled as generated data to maintain traceability.

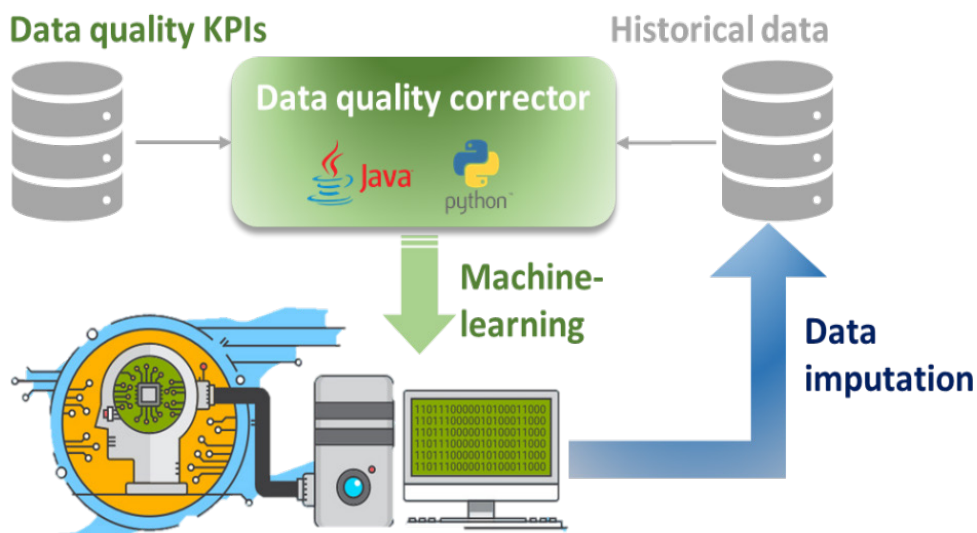


Figure 18: Data quality corrector based on machine-learning

3.5 Conclusions

Overall, the ETL mechanisms provide the practical backbone for turning heterogeneous TEF data into harmonised, experiment-ready datasets. It explicitly recognises different integration patterns (FTP-based and non-FTP sources) and adopts pragmatic engineering choices, such as the use of an intermediate FTP server, to reduce connection complexity and streamline data collection through consistent access paths.

Beyond ingestion, the ETL design is centred on repeatable and maintainable pipelines that clearly separate orchestration from transformation logic. This enables systematic filtering, cleaning, harmonisation, and enrichment of incoming data into a common EnergyGuard data model before loading it into the Data Lake. By embedding validation, quality checks, unit and metadata alignment, and temporal consistency directly into the pipelines, the ETL layer ensures that downstream digital twin and AI workflows are reliable, comparable, and scalable across different TEFs.

4 EnergyGuard ontology and semantics and alignment with data spaces and AIoD

4.1 Purpose and Scope

The EnergyGuard project supports multiple, heterogeneous energy sectors, including distributed generation assets, transmission and distribution networks, and varying classes of prosumer sites. Each sector employs distinct data schemas, terminology, and measurement protocols, resulting in a high degree of data heterogeneity across the consortium. To handle this, it is important to create a formal structural ontology and semantic data model to harmonize all internal related data. The ontology for EnergyGuard aims to provide a coherent, machine-interpretable representation of the various entities, relationships, and conceptual domains involved in the project. Its primary focus is to capture the structure of both physical and virtual energy systems, as well as the interactions between them. This framework ensures that all data collected across pilots and subsystems can be semantically harmonized, enabling consistent reasoning, analysis, and integration with other system components.

The integration with European Data Spaces (EDS) is also important to ensure the external relevance and trust of the EnergyGuard platform. A dedicated data space connection ensures secure data exchange and maintains compliance with governance models like GDPR, which is critical for protecting sensitive energy data. By adhering to standardized protocols such as those supported by the EDS, the project facilitates cross-sector interoperability, allowing seamless communication between EnergyGuard systems, IoT devices, and external platforms. This framework further builds transparency and trust by securing data provenance and providing a scalable, flexible infrastructure that grows efficiently as new systems are integrated.

Finally, a dedicated focus on alignment with the AI-on-Demand (AIoD) platform is necessary to maximize the usability of the semantically enriched data. This integration guarantees that the unified dataset and the resulting AI services are structured specifically to be consumable by AI/ML algorithms. This specialized formatting reduces friction in the development, deployment, and adoption lifecycle of EnergyGuard's intelligent solutions by external developers and end-users, ensuring that the project's outputs are immediately actionable in practical, real-world energy applications.

4.2 Conceptual Structure of the Ontology

The ontology covers a range of entities and relationships across multiple conceptual domains. Physical entities include devices such as PV panels, wind turbines, battery systems, diesel generators, pumps, and buildings, as well as grids that distribute energy. Virtual entities represent digital information derived from physical entities. This includes time-stamped measurements from sensors (e.g., PV production, wind output, battery charge/discharge), and energy services derived from operational data. These allow the system to monitor, analyse, and optimize the behaviour of the physical infrastructure. Relationships in the ontology capture measurements, control actions, ownership, location, and functional dependencies between system components. Together, these elements provide a comprehensive representation of the operational, structural, and functional aspects of energy systems within the pilots.

The ontology is structured according to a modular architecture. This allows it to evolve over time, supporting additional experiments, or new energy technologies, while ensuring that the core remains stable and interoperable. Modules are defined along three main axes:

- 1) Pilot-based modularization: Each pilot's infrastructure and data sources are represented in dedicated modules, allowing pilot-specific modelling without affecting the core ontology.
- 2) Domain-based modularization: Entities and relationships are grouped by conceptual domain—production, storage, consumption, and monitoring—which enhances maintainability and readability.
- 3) Physical versus virtual separation: Physical vs. virtual separation: Physical systems and their virtual counterparts are modelled in separate but linked modules. Physical modules describe the actual systems (e.g., electrolyzers, Pump, Valve etc), while virtual modules represent their sensor data (e.g., temperature, pressure, level etc). This clear separation allows the ontology to model both the real-world infrastructure and its digital representation, while maintaining semantic consistency.

4.2.1 Reference Ontologies and Standards

The EnergyGuard ontology builds upon existing, widely used ontologies and standards to ensure semantic interoperability, consistent measurement semantics, and reusability of published models. Table 4 presents the core ontologies and standards currently applied across pilots. It is not exhaustive: individual pilots may introduce additional domain-specific vocabularies, and future extensions may incorporate new standards as required by evolving datasets and experiments.

Table 4: Primary Semantic Standards and Vocabularies for EnergyGuard

Standard / Vocabulary	Purpose	Usage value
Semantic Sensor Network Ontology (SOSA) ¹	Provides a generic, domain-independent foundation for representing sensors, observations, data acquisition processes, sampling frequencies, features of interest, results, and actuation. Enables a consistent, machine-interpretable structure for time-series data across all pilots.	Measurement rows with timestamps and values. Attach sensors, features of interest, and deployment metadata via SSN where needed.
Smart Applications REference (SAREF) ²	Provides a consistent representation of devices, their functions, services, and energy-related capabilities. Covers appliances, meters, actuators, and energy systems. Supports alignment with EU smart home and smart grid initiatives.	Device classification, functional capabilities, and service roles.
SAREF4ENER (Energy)	Provides energy-specific concepts within SAREF, including energy flows, energy profiles, storage behaviour, power	PV systems, batteries, consumption/production flows, flexibility behaviour.

¹ <https://www.w3.org/TR/vocab-ssn/#intro>

² <https://saref.etsi.org/>

Extension of SAREF) ³	generation, and demand response. Bridges SAREF device semantics with smart energy domain requirements.	
DCAT (Data Catalog Vocabulary)	Provides a standard for describing datasets, distributions, publishers, formats, provenance, and metadata catalogues.	Cataloguing pilot datasets and describing ETL outputs.
QUDT (Quantities, Units, Dimensions & Types) ⁴	Offers authoritative, machine-interpretable unit and quantity semantics. Ensures consistency across pilots and prevents ambiguities such as "kW vs kW_total vs KW". Supports unit conversion and validation.	Unit properties of every numerical measurement.
Brick Schema ⁵	Comprehensive building/system topology covering HVAC, zones, equipment, and their sensors.	sensor placement, HVAC equipment, meters, and control points
BOT (Building Topology Ontology) ⁶	Canonical representation of building structure, spaces, zones, and building elements.	Spatial hierarchy, geometry of buildings, floors, spaces, and building components. Import BOT in digital twin models for spatial reasoning.
REC (Real Estate Core) ⁷	Standardized ontology for real estate assets, buildings, units, spaces, ownership, contracts, and metadata.	Energy audits, renovations, EPCs.
FOAF (Friend of a Friend) ⁸	Provides lightweight modelling of agents, including users, organizations, and groups. Useful for human-centric entities not covered by energy ontologies.	Community members, prosumers, aggregators or REC governance entities.

Figure 19 illustrates how the EnergyGuard approach is applied in Pilot 2 to transform a raw, column-based dataset into a semantically organised knowledge structure. The dataset contains timestamp fields alongside measurements from wind turbines, PV systems, batteries, pumps, thermal loads, and hydro resources. Rather than modelling these as disconnected spreadsheet columns, the ontology groups them into broad system-component categories (e.g., wind, solar, storage, thermal consumption, and hydro etc) each represented as a dedicated class. A dataset instance then links together the component groups it contains and provides high-level metadata such as file name, sampling interval, and temporal coverage.

³ <https://saref.etsi.org/saref4ener/v2.1.1/>

⁴ https://www.qudt.org/doc/2025/01/DOC_SCHEMA-QUDT.html

⁵ <https://brickschema.org/>

⁶ <https://w3c-lbd-cg.github.io/bot/>

⁷ <https://github.com/RealEstateCore/rec>

⁸ <https://sparontologies.github.io/foaf/current/foaf.html>

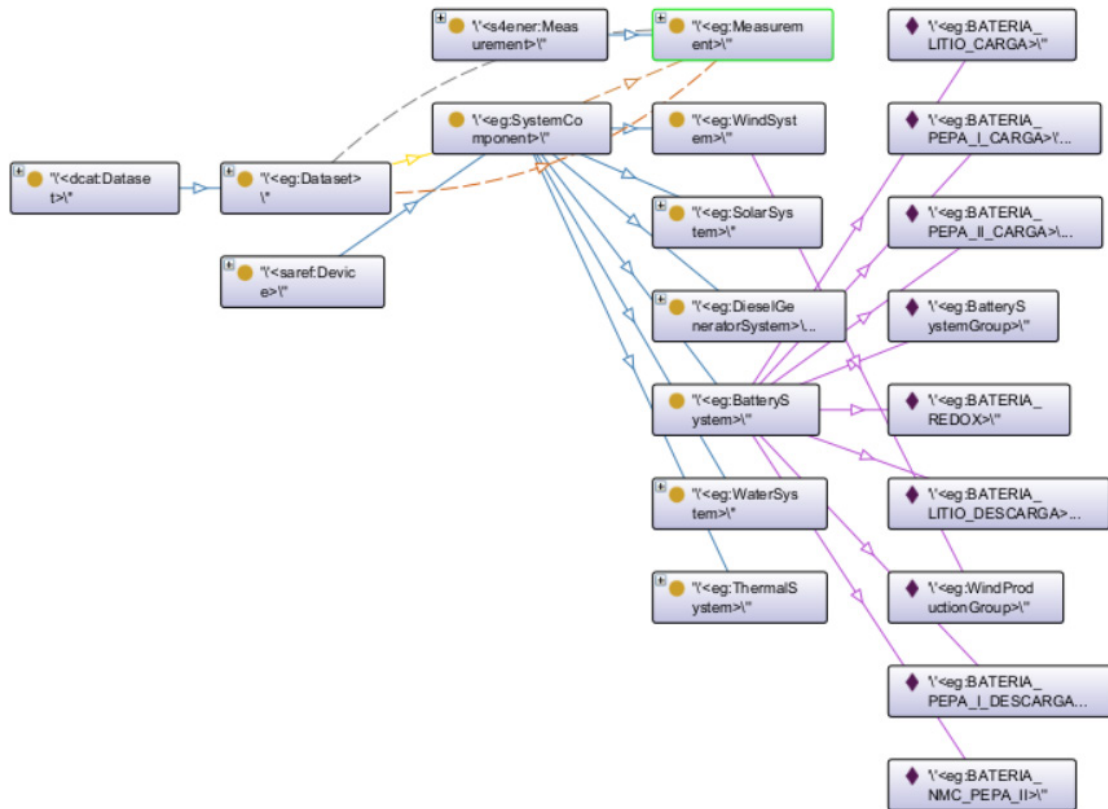


Figure 19: Pilot 2 initial data ontology schema

4.2.2 Conceptual Integration with EnergyGuard Components

Within EnergyGuard, the ontology acts as a stable reference model and vocabulary that describes the structure, systems, and data characteristics of each pilot. It is connected to the Data Lake through a synchronisation mechanism, allowing the Data Lake to use the ontology as a semantic reference when evaluating any new incoming data. In this way, each new data stream can be checked for compliance with the predefined pilot schema, ensuring consistency, quality, and interoperability across all datasets. At the same time, when new features, measurements, or subsystems emerge that were not previously represented, the ontology can be updated accordingly, maintaining an evolving yet well-controlled knowledge layer aligned with the real-world development of each pilot.

4.3 Integration with European Data Spaces

Integration with European Data Spaces is a key enabler for EnergyGuard, ensuring secure, interoperable, and scalable energy data exchange across heterogeneous stakeholders and platforms. Data spaces provide a structured framework for cross-organizational collaboration while maintaining data sovereignty, and strict compliance with EU regulations such as the GDPR. This is particularly important in the energy domain, where sensitive data (e.g., consumption patterns, asset performance, or grid status) must be protected through secure access control, auditing, and sovereignty-preserving mechanisms.

Beyond security, data spaces also play a pivotal role in enabling interoperability. Through widely adopted standards and protocols, such as NGS-LD, GAIA-X federation services, and IDS-based connectors, energy systems can seamlessly communicate with IoT infrastructures, smart homes, city

platforms, and industrial systems. This cross-sector interoperability supports unified data models and facilitates the integration of diverse datasets across the energy value chain. This integration allows EnergyGuard to leverage external datasets and AI services, supporting advanced analytics, predictive modelling, and seamless data exchange.

A key benefit of adopting data spaces is the enhancement of data transparency and trust. Built-in capabilities such as data provenance tracking, verifiable credential management, and auditing allow stakeholders to reliably trace data origins and usage. Furthermore, smart contracts or policy enforcement mechanisms can automate data-sharing agreements, ensuring that all energy data exchanges remain trustworthy, compliant, and aligned with predefined rules.

Data spaces also offer scalable and flexible infrastructure, supporting cloud-to-edge federation for low-latency, real-time processing as EnergyGuard expands its ecosystem. This environment fosters innovation and collaboration, enabling stakeholders to integrate advanced machine learning technologies for enhanced energy management and smart grid applications.

Finally, the adoption of a shared data exchange framework brings measurable cost-efficiency, by using a shared and standardized data exchange framework for all stakeholders.

The candidate connectors identified in D2.2 [1] (Eclipse Dataspace Connector, Sovity EDC, and TNO Security Gateway) provide the technical foundation for controlled data sharing, policy enforcement, identity and access management, and secure traceable exchanges. For a detailed discussion of their functionalities and planned use within EnergyGuard, see D2.2.

Since D2.2, further research has been conducted on European energy data space initiatives and connector adoption. Priority was given at examining the data space technologies and providers utilized by other relevant European projects to understand prevailing architectural choices and integration patterns. Table 5 summarizes the most relevant open-source energy dataspace connectors of the investigation.

Table 5: Overview of Energy Dataspace Connectors

Connector	Maintainer	TRL	Protocol	Related Energy Dataspace Projects
Eclipse Dataspace Connector (EDC) ⁹	Eclipse Foundation	8-9	Dataspace Protocol 2024-01 (HTTPS)	OMEGA-X, DATA CELLAR, energy data-X
FIWARE Data Space Connector ¹⁰	FIWARE Foundation	6-7	HTTPS (NGSI-LD)	Enershare (DOVE 4.0, i4Trust)
TNO Security Gateway (TSG) connector ¹¹	TNO	8	IDS Multipart	Enershare

⁹ <https://github.com/eclipse-edc/Connector>

¹⁰ <https://github.com/FIWARE/data-space-connector>

¹¹ <https://gitlab.com/tno-tsg/projects/enershare>

TRUE Connector ¹²	Engineering Ingegneria Informatica	6	IDS Multipart, IDSCP	InterSTORE, OneNet
Sovity EDC Connector ¹³	Sovity GmbH	9	Dataspace Protocol 0.8/1.0 (HTTPS)	Omega-X
Interstore Connector ¹⁴	OneNet / InterSTORE	7-8	IDS/OneNet	InterSTORE

At this stage, the landscape of energy data space technologies and providers is being investigated. The analysis of existing connectors highlights a diversity of maturity levels (TRL 6–9) and supported protocols. This initial assessment provides insight on how these solutions overlap in terms of technology, interoperability, and adoption in energy-focused projects (e.g., OMEGA-X, DATA CELLAR, Enershare, InterSTORE). Rather than committing to a single connector at this stage, multiple options are being explored to identify the best fit for EnergyGuard’s use case. This approach will enable a future-proof, interoperable energy data space architecture aligned with the evolving ecosystem of energy data providers and standards.

4.4 Integration with AI-on-Demand platform

The AI-on-Demand platform is a central digital hub that drives European AI research and innovation, while upholding excellence, trust, and transparency. EnergyGuard will consider AI-on-Demand’s architecture while developing its own, as mentioned in D2.2 [1]. Apart from this, EnergyGuard will perform integration with the AI-on-Demand platform by leveraging the following services:

- AI-Builder, a Graphene-based application that allows users to seamlessly orchestrate containerised AI solutions through an intuitive graph-based interface.
- AI-on-demand AI Marketplace, a pan-european AI marketplace where users can acquire and deploy AI related products.

More specifically, EnergyGuard will offer its assets with the greatest business and market potential as services in the Marketplace. In that way, these solutions will become easily discoverable by a broad range of energy stakeholders, maximising their reach, fostering real-world adoption, and paving the way for sustainable commercial exploitation beyond the project’s lifetime. Also, AI-Builder will be utilised to publish selected AI models developed within EnergyGuard. This will allow their integration into real-world workflows, enable seamless deployment and composition with other AI services deployed on AI-Builder, and enhance their visibility. Through this close architectural alignment and technical integration, EnergyGuard will benefit from and enrich the AI-on-Demand platform, contributing valuable energy AI services and models by enabling a launch pipeline from EnergyGuard TEF to AIOD marketplace through its product onboarding process currently under development (as of December 2025)

¹² <https://github.com/Horizont-Europe-Interstore/Data-Space-Connector/>

¹³ <https://github.com/sovity/edc-ce>

¹⁴ <https://github.com/Horizont-Europe-Interstore/Data-Space-Connector>

4.5 Conclusions

In conclusion, the EnergyGuard ontology acts as the semantic backbone of the platform, reducing cross-pilot heterogeneity by providing a shared, machine-interpretable model of entities, measurements, and relationships in energy systems. Given the diversity of pilots, technologies, and data formats, this semantic layer is essential for establishing a common understanding that supports consistent integration, reasoning, and reuse of data across the ecosystem.

The ontology is designed to be operational rather than static, remaining synchronised with the data lake so that incoming datasets can be validated against agreed schemas while still allowing controlled evolution as new assets or measurements emerge. At the same time, its alignment with European Data Space principles and AI-on-Demand requirements ensures that semantically enriched data and services can be shared securely, interoperable, and in a form that is readily consumable by AI and analytics workflows beyond the immediate platform.

5 Data sharing via intelligent query engine

In a context of energy transition and increasing digitalization, players in the energy sector are facing an explosion in data volumes. This data comes from multiple sources: IoT sensors, smart meters, weather forecasts, consumption histories, maintenance data, energy markets, etc. In this section, we focus specifically on the management of this data in a data lake and its access through an intelligent query engine for which we will present our selection criteria and implementation options. This preliminary knowledge was obtained by compiling the following open-source information (mainly smart grid oriented) who also provided us with material for testing our first developments:

- « Electricity Transmission and Distribution Report and Database » [7]
- « Derived Map of Global Electricity Transmission and Distribution Lines » [8]
- « Big data analytics in power distribution systems » [9]

5.1 Issues in Energy Management

Modern energy management is facing a profound transformation accompanied by major challenges linked to data and this mutation is becoming a strategic lever but also a source of complexity. Data management issues are listed and illustrated by the smart grid application as follows:

5.1.1 Multiplicity of data sources

Current energy systems generate a massive amount of data from diverse sources: Production (e.g. thermal power plants, wind turbines, solar panels ...), Distribution (e.g. smart grids, IoT sensors ...), Consumption (e.g. smart meters, connected buildings, electric vehicles, ...) and External sources(e.g. weather data, energy markets, public policies...).

This diversity creates challenges in terms of integration, standardization, and reliability. Formats vary, update frequencies differ, and volumes are growing exponentially. Without an appropriate architecture, this data remains underutilized.

5.1.2 Complexity of decisions

Decisions in the energy sector are simultaneously becoming increasingly complex in the following aspects:

- Multi-criteria: Necessity to take into account technical, economic, environmental and social parameters.
- Real-time: Production or distribution adjustments must be responsive to fluctuation in demand or climate hazards.
- Anticipation: Consumption must be forecast, investments optimized, and risks managed.
- Coordination: There are many stakeholders (producers, distributors, regulators, consumers) and their interests sometimes diverge.

This complexity requires powerful decision-making tools capable of handling dynamic and uncertain scenarios.

5.1.3 Need for rapid and relevant data interrogation

In this context, access to data can no longer be a barrier with additional requirements such as:

- Speed: Operators must be able to query data in seconds to act effectively.

- Relevance: Information must be contextualized, filtered, and presented in an understandable manner
- Interoperability: Systems must communicate with each other to avoid silos.
- Security: Sensitive data must be protected against intrusion and alteration.

Intelligent access to data is hence becoming a key factor in the performance, innovation, and resilience of energy systems.

5.2 Intelligent Query Engine

To take full advantage of the above considerations, it becomes imperative to have an infrastructure capable of centralizing, structuring and efficiently querying this information. This is where the data lake and the intelligent query engine come in.

A data lake allows the ingest of heterogeneous data—structured, semi-structured, or unstructured—at scale. Unlike traditional databases, it offers maximum flexibility for integrating data streams in real time or in batch mode, without prior transformation.

5.2.1 Role of the Intelligent Query Engine

An intelligent query engine allows for fast, relevant, and contextual querying of the data lake. It is a distributed query engine capable of querying data stored across multiple heterogeneous sources (relational databases, NoSQL, data lakes, etc.) without moving it. It acts as an intelligent abstraction layer that enable:

- Unified access to distributed data
- Query optimization based on source
- Parallel and distributed execution
- Federation of queries across multiple systems

These engines are essential in modern data architectures, particularly in Data Mesh, Lakehouse, or multi-cloud environments.

It relies on advanced technologies such as natural language processing (NLP), semantic indexing, and query performance optimization. Its benefits for energy management are numerous

- Predictive analytics: anticipate consumption peaks or outages using complex correlations between historical data and current conditions.
- Real-time optimization: adjust production or distribution based on needs, market prices, or technical constraints
- Augmented decision-making: provide operators with recommendations based on intelligent queries, even without data science expertise.
- Anomaly detection: automatically identify unusual behaviour in networks or equipment.

5.2.2 Investigation of Intelligent Query Engines (IQEs)

A quick analysis reveals the strengths of IQEs in terms of Federated access (i.e. query multiple sources without ETL), Performance (e.g. distributed and parallelized execution), Flexibility (e.g. support for a variety of formats such as Parquet, CSV, JSON, etc.) and Interoperability (e.g. connectors to HDFS, PostgreSQL, Kafka, MongoDB, etc.), but also weaknesses such as No persistence (i.e. these are query engines, not databases), Configuration complexity especially in a distributed environment, sensitive to

network latency, especially in federated mode and Security and governance which must be managed externally.

The table below briefly presents the results of our analysis of existing IQEs on the market and from this analysis, we selected Trino as the basic IQE to support further our developments.

Table 6: Summary of the analysis of the existing IQEs

Tool	Description	Strengths	Limits
Trino	Open-source distributed SQL engine, formerly PrestoSQL2	- Federated queries - High performance - Broad connectivity	- No storage engine - Network dependency
PrestoDB	Presto variant developed by Facebook	- Fast for interactive queries - Hadoop integration	Less active than Trino on the community side
Apache Drill	Schema-free engine, designed for semi-structured data	- Flexible on formats - No need to define a schema	Less efficient on large volumes

5.2.3 The Intelligent Query Engine Trino

Trino [10] was created by the original developers of Presto at Facebook. In 2019, they forked the project as PrestoSQL, which was renamed Trino in 2020. Today, Trino is maintained by the Trino Software Foundation and widely adopted by enterprises and cloud platforms, including Amazon Athena and Starburst Galaxy.

Trino operates on a cluster composed of a Coordinator responsible of queries parsing/optimization/scheduling and Workers to execute distributed tasks. Trino is Java-based and SQL compatibility, i.e. it supports ANSI SQL, making it compatible with BI tools like Tableau, Power BI, and Apache Superset.

Trino excels at querying multiple data sources simultaneously through its connector-based architecture. These connectors act as bridges between Trino and various storage systems or databases: Relational databases (e.g. MySQL, PostgreSQL, ...), NoSQL systems (e.g. MongoDB), Data lakes (e.g. HDFS, AWS S3, Google Cloud Storage, Azure Blob Storage ...) and File formats: (e.g. CSV, JSON, Parquet...). This enables federated queries, allowing users to combine data from different systems in a single SQL statement.

5.3 Work Plan

The main steps of our approach would be discovering the issues of using IQEs in data management in the energy sector (explicit in the previous paragraphs of this section), the implementation and testing of the IQE Trino in single-server mode followed by its porting and optimization on the Meluxina HPC platform and finally its integration into the project environment to support the operation of the different TEFs.

5.3.1 Single-Server Mode

During the first phase of our development, we are focusing on implementing the core tools in a single-server environment. These tools currently include

- SQLite in the Julia programming environment

- Relational database MySQL
- Trino in single-server mode.

These implementations will be validated by tests on the generic smart grid application as illustrated by an example in a subsequent paragraph.

5.3.2 Implementing IQE in an HPC environment

Integrating an IQE like Trino into an HPC environment would bring power to the operation of large data lakes but requires several adjustments. Indeed, Trino operates according to a master-slave architecture and in an HPC environment, this architecture can be mapped as follows:

- Coordinator: Master or front-end node to serve as entry point for SQL queries, schedule and orchestrate execution.
- Workers: Use HPC compute nodes as Trino workers to execute distributed tasks, read data, and return results.
- Distributed storage: Connect to HDFS or Lustre for data (should be mounted on all nodes)
- Orchestration: Integrate with Slurm or Kubernetes for resource and jobs management
- Network optimization: Prioritize high-speed interconnections (InfiniBand)

5.4 Testing on examples of smart grid

5.4.1 Purposes of the tests

We provide an illustration of IQE by application to the smart grid use case in this section: The integration of an Intelligent Query Engine (IQE) into smart grid infrastructure revolutionizes how energy systems manage, analyse, and respond to vast streams of real-time data. In this context, the IQE acts as a dynamic question-answering system that enables operators, engineers, and consumers to interact with complex grid data.

While waiting for the integration with the EnergyGuard TEFs, we will use snippets of data gleaned from the Internet to support these tests.

By leveraging advanced retrieval-augmented generation (RAG) techniques, the IQE can process heterogeneous datasets—such as sensor readings, meter logs, and control parameters—and deliver context-aware, accurate responses.

In a first step of testing, we will explore lightweight, embedded database solutions for edge computing or local data aggregation. This allows us to test the tool implementation while learning about the demands and requirements of this specific application, the smart grid.

5.4.2 Illustration of tests on an example of a smart grid

- Objective: Simulate a smart grid environment with multiple data sources.
- Data sources may include Smart meters (energy consumption per household), Weather stations (temperature, humidity, solar radiation), Grid sensors (voltage, frequency, fault detection), External APIs (energy pricing, demand forecasts).

Examples:

Meter_Data (6730×5 DataFrame)					
Row	Time Stamp	Household_Id	kWh	Voltage	Current
1	01/10/25 00:00	1	0.2	224.5	6.6
2	01/10/25 00:00	2	1.1	221.0	9.4
3	01/10/25 00:00	3	1.6	233.5	5.3
... etc.					

Weather_Data (169×4 DataFrame)					
Row	Time Stamp	Temperature	Humidity	Solar_irradiance	...
1	01/10/25 00:00	16.0	31.0	580.0	...
2	01/10/25 01:00	24.5	48.0	770.0	...
3	01/10/25 02:00	12.0	77.0	60.0	...
... etc.					

Pricing Data (169×2 DataFrame)		
Row	Time Stamp	Price/kWh
1	01/10/25 00:00	0.24
2	01/10/25 01:00	0.1
3	01/10/25 02:00	0.14
... etc.		

Operating scenario:

- Simulate smart meters collecting household energy consumption, e.g. Query average daily consumption per household, Identify peak usage hours across all households...

Example :

```
SELECT household_id, DATE(timestamp) as day, ROUND(AVG(kWh), 2) as avg_kWh  
FROM Meter_Readings GROUP BY household_id, day ORDER BY household_id, day;
```

Average daily Consumption (80x3 DataFrame)			
Row	Household_Id	Day	Avg_kWh
1	1	2025-10-01	1.03
2	1	2025-10-02	0.91
3	1	2025-10-03	1.11
... etc.			

- Correlate weather data with energy demand to predict consumption including join smart meter readings with weather data by matching timestamps, Compute the correlation between temperature and energy consumption...

Examples:

Join Meter and Weather Data (6730x6 DataFrame)						
Row	Time Stamp	Household_Id	kWh	Temperature	Humidity	Solar_Radiance
1	01/10/25 00:00	1	0.2	16.0	31.0	580.0
2	01/10/25 00:00	2	1.1	16.0	31.0	580.0
3	01/10/25 00:00	3	1.6	16.0	31.0	580.0
... etc.						

Join Meter and Pricing Data (6730x5 DataFrame)					
Row	Time Stamp	Household_Id	kWh	Price/kWh	Cost
1	01/10/25 00:00	1	0.2	0.27	0.05
2	01/10/25 00:00	2	1.1	0.27	0.30
3	01/10/25 00:00	3	1.6	0.27	0.43
4	01/10/25 00:00	4	2.0	0.27	0.54
... etc.					

Correlation between temperature and energy consumption: -0.009

- Detect Anomalies

Detection of Energy Anomalies (Z-Score) using a simple rule-based mechanism

Compute z-scores for kWh

6730-element Vector{Float64}:
 -1.4636087261927944
 0.0926237068011064
 0.9571972806866067
 1.6488561397950068

...

Detection of Pricing Anomalies (1682×5 DataFrame)					
Row	Time Stamp	Household_Id	kWh	Price/kWh	Cost
1	01/10/25 00:00	3	1.6	0.27	0.43
2	01/10/25 00:00	4	2.0	0.27	0.54
...					
21	01/10/25 02:00	2	1.9	0.16	0.30
...					
31	01/10/25 03:00	2	1.8	0.13	0.23
... etc.					

Plots

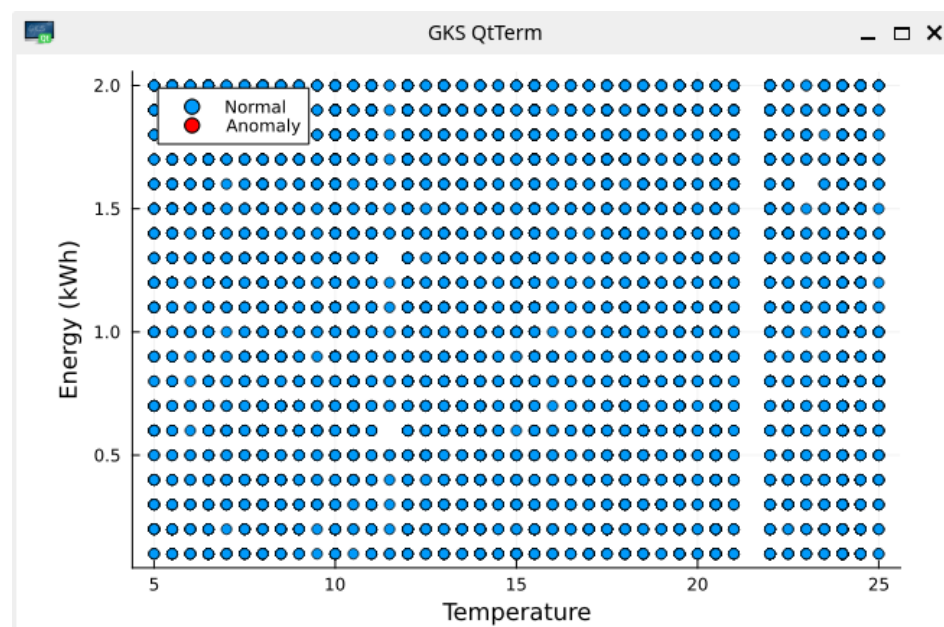


Figure 20: Energy anomaly

5.5 Conclusions

The Intelligent Query Engine enables unified, high-performance access to distributed and heterogeneous data sources, transforming the data lake into an actively usable asset for

experimentation and operations. By providing a federated query layer, it allows users and services to interrogate data in place, reducing unnecessary data duplication while supporting complex analytical queries across multiple systems.

The design explicitly balances benefits and constraints. While federated query engines offer scalability, interoperability, and parallel execution, they also introduce operational complexity, dependency on network performance, and limitations such as the lack of native persistence. The staged implementation strategy, progressing from single-node validation to HPC-enabled deployment and full TEF integration, ensures that performance, robustness, and operational feasibility are addressed incrementally, aligning the technology choice with the realities of large-scale energy experimentation environments.

6 Digital twins configuration and adaptation

This chapter introduces the process through which the project will analyse, adapt, and configure the existing Digital Twins (DTs) from the TEFs to ensure their full usability within the EnergyGuard framework. The initial step involves assessing the capabilities, structure, and maturity of each local DT to verify how well they meet the functional and technical requirements of EnergyGuard. These DTs typically represent both the physical and virtual infrastructures of the TEFs, exposing virtual assets, data models, inference interfaces, software-as-a-service APIs, user-facing dashboards, and other operational components.

A key objective of this work is to ensure interoperability across the different TEFs. To achieve this, dedicated integration mechanisms, through well-defined APIs, will be designed and implemented so that each DT can be seamlessly connected to the overarching TEF Foundation Framework. This common interface layer will enable consistent access to data, simulation resources, and AI services across all facilities.

Once adapted, the DTs will be configured according to the needs of each experimentation scenario and the specific AI-based services to be tested. The configuration process will follow the AI testing pipeline provided by the Service Execution Manager, ensuring that all DTs operate under a harmonised workflow. This includes defining the required input data (ranging from batch or historical records to synthetic datasets or simulations produced by physical Building Energy Models) as well as the outputs expected from each experiment. Through these configurations, the DTs will become a core operational component, enabling controlled, repeatable, and interoperable experimentation within EnergyGuard.

6.1 Interoperability API definition

To enable secure, standardised, and efficient data exchange between the central PostgreSQL database hosted in the Meluxina HPC environment and the external TEFs participating in EnergyGuard, a dedicated access API will be designed and implemented. This API will act as the controlled interface through which authorised external entities can query, retrieve, and interact with the data processed and stored by the ETL pipelines.

The API will expose curated datasets and processed outputs generated by the ETL modules, ensuring that external TEFs, regardless of their internal communication capabilities, can seamlessly access the information required for their experimentation workflows. By abstracting the underlying data structures and internal HPC processes, the API provides a unified and consistent data access layer while preserving security, integrity, and performance of the Meluxina environment.

Figure 21 shows the schema where the API would be established. There exist two ways to generate this communication, which is under definition in this first release:

- Option 1: Poll requests from the DTs for data. This would imply the creation of an endpoint within the virtual machine hosting the database in order to provide a query access to extract data.
- Option 2: Push messages where a script would extract data from the database to push data periodically to the DTs.

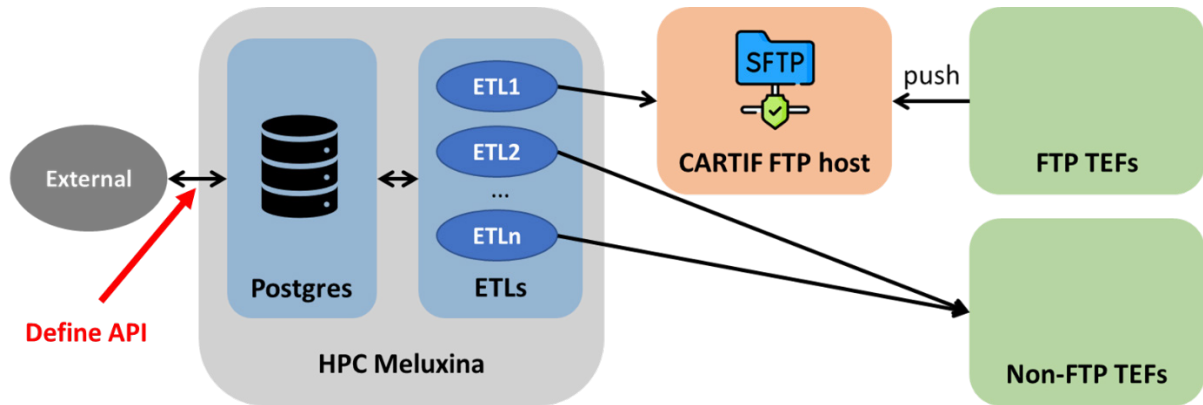


Figure 21: API level definition for the communication with the DTs

The key functional objectives of the API and adaptation of the DTs are described in the next bullets:

- Standardised data exposure to expose data stored in PostgreSQL in a format independent of the internal schema. This ensures compatibility across TEFs and simplifies integration with external systems.
- Secure external access, providing the authentication and role-based authorisation to ensure that only validated TEFs and partners can request datasets. Data transfer will be encrypted, complying with HPC security policies and GDPR requirements when applicable (see section 8 for more details about privacy and security).
- Support for heterogeneous TEFs, enabling harmonised data delivery to both FTP-enabled and non-FTP TEFs.
- Efficient data retrieval as endpoints will allow filtered queries based on experiment, time periods, data types, asset identifiers, or scenario-specific parameters. To optimise performance, the API will incorporate pagination, asynchronous processing for heavy requests, and caching mechanisms when appropriate.
- Metadata and data model transparency, including the description of the datasets, update timestamps, data lineage, and formats. This ensures external users can correctly interpret the data and maintain consistency across experiment runs.
- Integration with the ETL layer as the API interacts closely with the ETL modules running inside the HPC environment. As these pipelines ingest, clean, transform, and enrich raw data, they ensure that only validated and harmonised datasets are exposed to the API. This guarantees that external consumers always receive high-quality, experiment-ready data aligned with EnergyGuard's requirements.

As output, the API will deliver a complete specification of available endpoints and accepted request / response formats (in JSON format).

6.2 Analysis of the DTs statuses

The EnergyGuard project involves a diverse set of Digital Twins (DTs) originating from different TEF infrastructures, each representing varying levels of maturity, data requirements, and customisation capabilities. Table 7 summarises the types of inputs and outputs handled by each DT, the formats used, the mechanisms available for configuration, and their exposed interfaces. Understanding these characteristics is essential to harmonising DT integration within the EnergyGuard framework. In short:

- **Portugal Transmission Network DT (RDN):** The Digital Twin representing the Portuguese high-voltage transmission grid consumes 15-minute resolution generation and consumption data through HTTP-based inputs. Its primary role is to produce simulation results concerning power system transients, which are exported in .csv format. Since this DT is already fully defined and operates at system scale, no configuration mechanisms are required for its internal parameters. Access to the DT is provided exclusively through the EnergyGuard Datalake, which acts as its main interface for data sharing.
- **CEDER-CIEMAT Microgrid and Distributed Energy Resources:** The microgrid DT integrates energy demand and generation prediction inputs, typically supplied either as .csv files or DataFrames. Based on these inputs, the DT computes the energy balance of the microgrid and exposes the results in .csv and/or JSON formats. Customisation is achieved using configuration files where scenario parameters can be defined by the user. Although the interface for external access is still to be determined, this DT already provides a flexible mechanism for adapting its behaviour to different prediction or operational conditions.
- **CEA LITEN Hydrogen Testing Platforms:** The hydrogen testing DTs at CEA LITEN rely on multiple input formats—.csv, .txt, and .mat (Matlab)—reflecting the heterogeneity of sensor data, operational parameters, or synthetic datasets used in hydrogen system modelling. The outputs include a wide range of system information such as sensor/actuator readings, system state, fault indicators, and hydrogen production rates, using the same formats as inputs. Customisation is enabled through parameterisation files and timeseries datasets that allow tailoring DT behaviour to specific user scenarios. Interfaces for this DT are still under definition.
- **BER Hydrogen Platforms:** For the hydrogen platform operated by BER, no Digital Twin instance is currently available. Consequently, no data formats, outputs, configuration capabilities, or interfaces have been defined at this stage.
- **CARTIF Hydrogen Testing Platforms:** The hydrogen DT developed by CARTIF uses several categories of inputs (i.e., PV generation, load demand, climate conditions, operation parameters, and user-specific parameters passed through configuration files). These are handled primarily using DataFrames for inputs and .csv or JSON formats for outputs. The DT generates performance metrics, energy flows, and set-points that are required by downstream control or analytical processes. Configuration is performed through parameter files, but no interface has yet been made available for external consumption.
- **Riga Multi-Apartment Residential Buildings:** The DT associated with Riga's residential building stock is driven by building-level information including address, building type, and energy efficiency measures selected by the user. Input/output formats remain to be defined, but the DT is designed to produce predictions of heat consumption, existing energy performance calculations, and the expected impacts of renovation measures. Outputs include energy savings, CO₂ reductions, financial savings, and post-renovation performance indicators. The DT is customised using variables such as building typology, audit results, renovation packages, and local data-driven AI models. Interfaces include the GeoRiga platform and the EFI-DSS system.
- **Antrodoco Renewable Energy Community:** No Digital Twin instance is currently available for the Antrodoco energy community, and therefore no associated data structures, configuration mechanisms, or interfaces have been established.

Table 7: DTs instances analysis from the TEFs

TEF name	Data inputs	Data Input Format	Data outputs	Data Output Format	Configuration mechanisms / How the DT is customised?	Available interfaces
1. Digital Twin (DT) of the Portuguese Transmission Network (RDN)	Transmission Grid generation and consumption 15-min data	HTTP	Simulation's transit-period result data	.csv	Not applicable	EnergyGuard Datalake
2. CEDER-CIEMAT Microgrid with its Distributed Energy Resources (DERs)	Energy demand and generation prediction	.csv or DataFrame	Energy balance	.csv and / or JSON	Config file with parameters	TBD
3. Hydrogen testing platforms at CEA LITEN	User scenario. The inputs accessible by the end-users are not yet defined and validated.	.csv, .txt, .mat (Matlab format)	All the usefull system informations (sensors and actuators values, state of the system, faults, H2 production rate...)	.csv, .txt, .mat (Matlab format)	The DT could be customized with parametrization files and timeseries inputs corresponding to the required scenario from the end-user.	TBD
3. Hydrogen testing platforms at BER	Not DT instance available	Not DT instance available	Not DT instance available	Not DT instance available	Not DT instance available	Not DT instance available
3. Hydrogen testing platforms at CEDER-CIEMAT	See TEF 2	See TEF 2	See TEF 2	See TEF 2	See TEF 2	See TEF 2
3. Hydrogen testing platforms at CARTIF	PV generation	DataFrame	Set-points	.csv and / or JSON	Config file with parameters	Not available
	Demand (load)		Performance metrics			
	Climate conditions		Energy flows			
	Operation parameters					

	User parameters	config file				
4. Riga's multi-apartment residential buildings (GEO RIGA, SCCB CIS)	Building address	TBD	Predicted heat consumption	TBD	Customised based on energy audit categories, renovation measures, building type, and energy use patterns using AI models trained on Riga data	Georiga
	Building type	TBD	Calculations of existing energy performance	TBD		EFI-DSS
	Energy efficiency measures could be selected by user		Energy efficiency measures			
			Heat energy savings			
			Renovation costs			
			CO2 savings			
			Euro savings			
			Predicted energy performance after renovation			
5. Antrodoco Renewable Energy Community (EnGreen)	Not DT instance available	Not DT instance available	Not DT instance available	Not DT instance available	Not DT instance available	Not DT instance available

6.3 Adaptation and customisation of the TEFs DTs

The adaptation of existing Digital Twins to the EnergyGuard framework follows a structured multi-stage workflow (illustrated in Figure 22) designed to harmonise heterogeneous DTs, integrate real system data, configure scenario-specific parameters, and produce simulation and optimisation outputs for end-users. The process ensures that diverse TEF infrastructures can be aligned with a common testing and experimentation methodology.

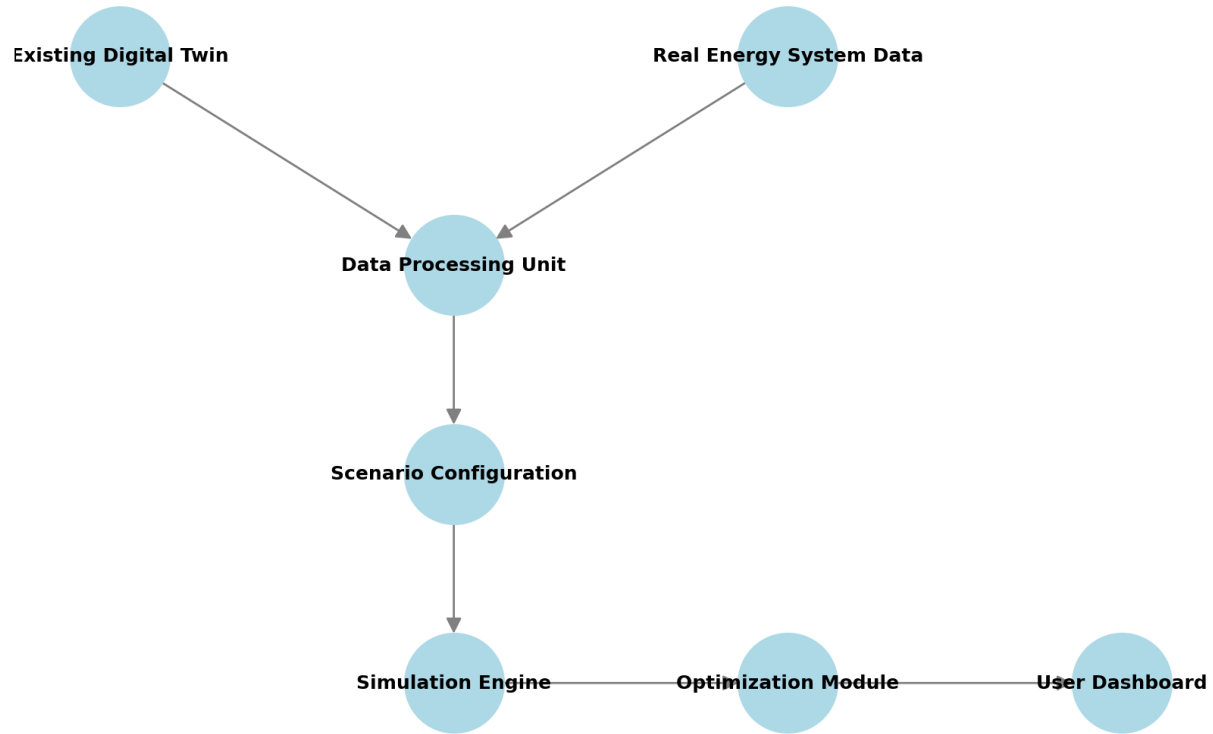


Figure 22: Workflow for the DTs adaptation and customisation

The main workflow stages are described below.

1. Integration of existing DTs and real energy system data: The workflow begins by incorporating two key sources of information:
 - a. Existing DTs models: These may include already-validated simulation models, historical datasets, ontologies, or operational logic representing the TEF infrastructure.
 - b. Real energy system data: This includes real-time or historical measurements such as load curves, generation profiles, weather data, operational states, or user-defined control parameters.

Both sources serve as foundational inputs needed to contextualise the DT and align its behaviour with current or expected system conditions.

2. Data Processing Unit that acts as the central harmonisation layer. Here, input data is cleaned, validated, transformed, and mapped to the internal data structures required by the DT and the EnergyGuard framework. Key responsibilities include:
 - a. Handling missing values, noise, or inconsistencies; temporal and spatial alignment (e.g., resampling to common time steps or aligning building/asset identifiers);
 - b. Feature extraction and enrichment, such as computing KPIs, operational states, load categories, or scenario metrics;

- c. Interfacing with ETL pipelines to ensure standardised processing across TEFs.
- 3. Scenario configuration: User-defined or experiment-defined scenarios are prepared. This step adapts the DT to the specific conditions under which AI services will be tested. Scenario configuration involves:
 - a. Selecting the type of experiment (e.g., forecasting, fault detection, optimisation, control).
 - b. Defining operational parameters, such as set-points, control rules, or boundary constraints.
 - c. Injecting user-defined settings, such as renovation measures, hydrogen demand profiles, or DER control strategies.
 - d. Incorporating time series inputs (e.g., predicted PV generation, heat demand, or synthetic datasets).
- 1. Loading configuration files that govern DT behaviour.
- 4. Simulation engine. Once scenarios are fully configured, the engine executes the DT model. This engine reproduces the physical or virtual behaviour of the energy system under the defined conditions. Simulation results form the core dataset used for AI evaluation, optimisation, or decision-support tasks.
- 5. Optimisation module. In scenarios where optimisation is required, the simulated outputs feed into an optimisation module, which applies AI or mathematical optimisation techniques to improve system performance.
- 6. User dashboard. Simulation and optimisation outputs are delivered to the user dashboard, where stakeholders can visualise, interpret, and compare results.

6.4 Conclusions

In conclusion, the digital twin configuration and adaptation approach is structured to bring heterogeneous digital twins under a coherent experimentation framework. By combining existing models with real system measurements, the approach ensures that digital twins remain context-aware and representative of actual or expected operating conditions, rather than remaining isolated simulation artefacts.

Central to this process is a harmonisation layer that handles data cleaning, validation, transformation, and feature enrichment, while managing temporal and spatial alignment. This enables consistent feeding of digital twins with high-quality inputs and supports reproducible experimentation. The workflow is completed through explicit scenario configuration, simulation and optimisation execution, and structured output delivery to user-facing dashboards, ensuring that digital twins are not only technically integrated but also operationally useful for evaluating AI-based energy services.

7 High-Performance Computing (HPC)

This section provides an overview of the High-Performance Computing (HPC) environment that underpins the execution and deployment of the EnergyGuard framework. Rather than introducing new functional requirements or standalone components, it describes the computational, storage, and networking context upon which the services and workflows defined in the other sections depend.

The HPC infrastructure is presented as an enabling layer that supports scalability, performance, and operational robustness for data-intensive and compute-intensive tasks such as large-scale data processing, simulation, optimisation, and AI model execution. By outlining the characteristics and constraints of the available HPC resources, this section establishes the boundary conditions that inform architectural and deployment decisions across the platform, ensuring that the functional components described elsewhere can be effectively instantiated and operated in a high-performance environment.

Figure 23 represents the key components of Meluxina.

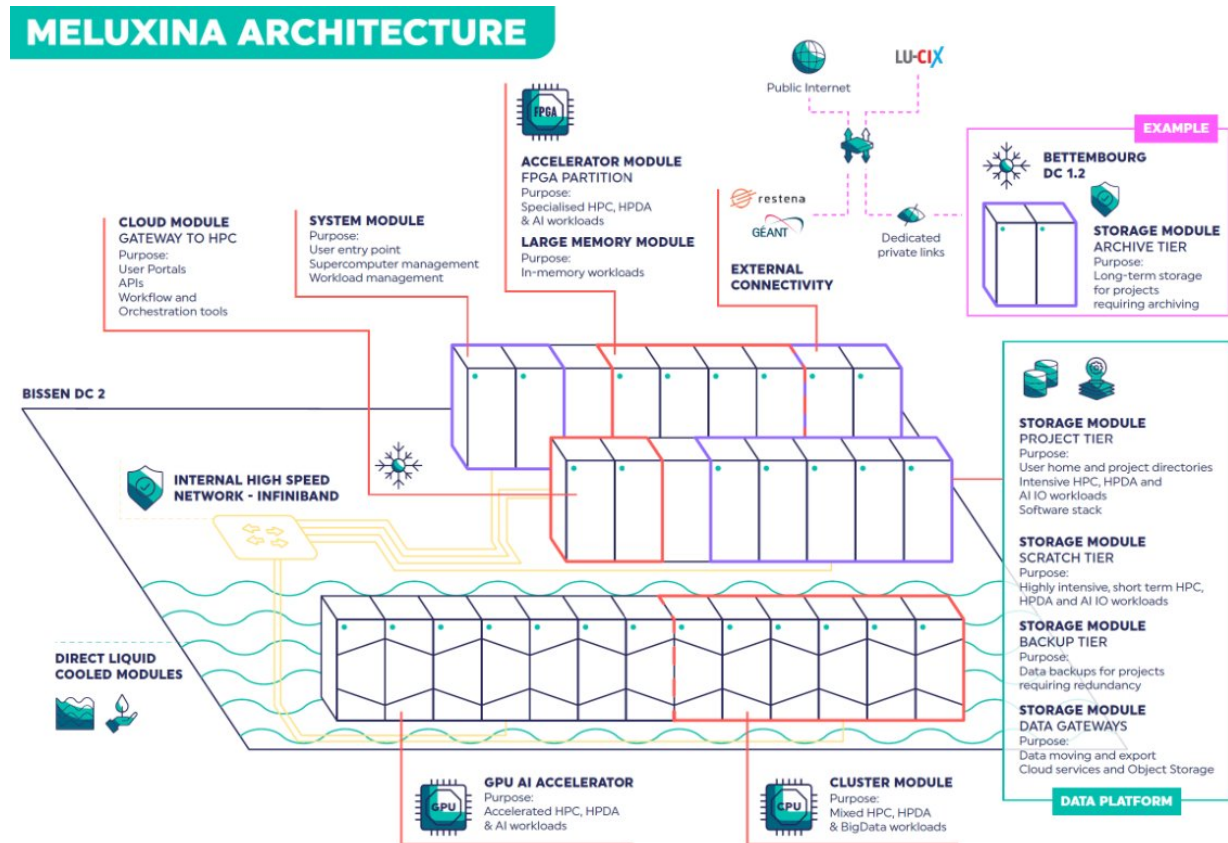


Figure 23: HPC Architecture (Meluxina)

The main components of the HPC architecture that are relevant to the data services are:

Storage

- Network file server for virtual machines, based on Ceph, from 64 to 256 GB.
- Network file server for HPC compute nodes and accessible to cloud VM, any size.
- Local, temporary, storage on GPU and large memory nodes of 1,92TB or 3,84TB.

Compute

- Cloud VM, from 2 to 64 vCPUs
- HPC compute nodes, either 128 or 256 cores, with 512GB or 4TB RAM

HPC accelerators, either Nvidia A100 or FPGANetworking

Cloud VMs (Virtual Machines) are accessible from the public internet through a VPN (Virtual Private Network) connection, such a connection can be permanent from a fixed IP address. Cloud VMs and HPC compute nodes can reach any public IP address. However, HPC compute nodes are not natively accessible from the public internet.

7.1 Mapping requirements to Meluxina resources

This section describes how the computational, storage, and networking requirements derived from the EnergyGuard framework are mapped onto the specific resources provided by the Meluxina HPC infrastructure. Building on the functional and non-functional requirements defined in previous sections, it explains how different categories of workloads can be aligned with the available execution environments, taking into account performance needs, scalability, persistence, and connectivity constraints.

7.1.1 Requirements on Meluxina resources

Storage wise, the overall required capacity is estimated to 150 GB for 2 years (from the sources identified). This amount of data can be stored either on cloud CEPH storage, or on the HPC network storage. The storage requirements represent the database sizes, estimated from the ingested data from the various sources.

Compute wise, the requirements are to support the database engines (Postgres, GraphDB, ...), the semantic search components (for example: Trino), the ETL processes, the ingestion process, the ETL processes including the data quality processor. Cloud VMs are largely capable of hosting the different servers. As a complement, it is usual to launch temporary computation on HPC nodes, for example if GPU cards are needed (Data quality processes are defined as machine learning based, which could be executed on regular VMs, but could also be launched on GPU nodes, for the duration of an execution). Also, some ETL processes can also be run on compute nodes, for a limited time, if they are compute-intensive (or benefit from GPU acceleration).

Network wise, the data ingestion is designed to pull data from the different sources. In case some sources require a push, CARTIF has planned an FTP server to stage the data, for later pull from the VMs or compute nodes (see Figure 24).

7.1.2 Overall design elements

From Figure 24 and initial design documents of the data lake and associated services, the following implementation notes apply:

1. Storage, given that the Lustre network storage can be available for the cloud VMs (via *sshfs* for example) in addition to the HPC nodes, the poor performance for database persistence of CEPH, and the large capacity of Lustre network file system, the recommended storage option is to use the HPC network file storage (Lustre) for all storage requirements. Also, the HPC network file storage is backed up. The VM will natively come with sufficient CEPH storage capacity in case

some further tests are needed. In case some data ingestion processes are launched on HPC nodes, the data will already be stored on the HPC storage.

2. Compute, the long-running services such as database server or workflow managers can be hosted on Meluxina Cloud partition (VMs). Several virtual machines can be dedicated to hosting these services. Their IP addresses then remain constant but require a VPN connection (a permanent link can be set up). Additional computational resources can be requested to execute compute, memory or storage intensive tasks. These additional resources belong to Meluxina's HPC partition. The tasks are not permanent, but batch oriented, and last as long as the tasks do. The tasks can be launched programmatically through a REST API.

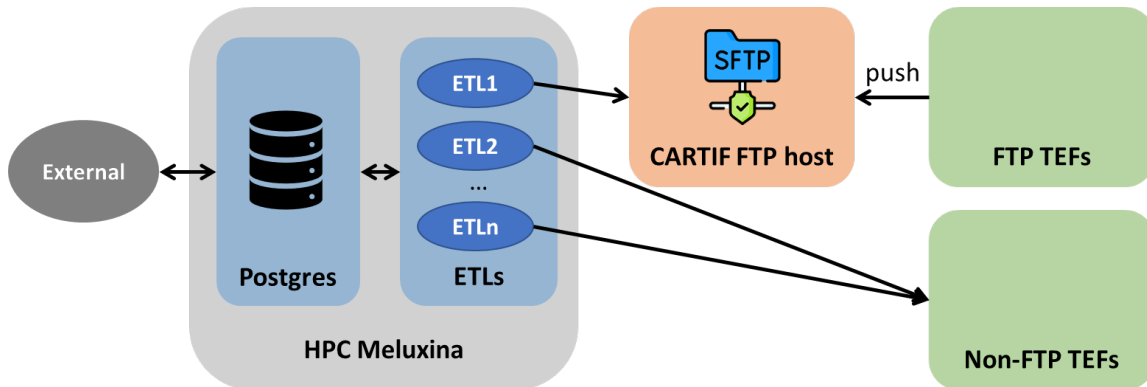


Figure 24: ETL and Meluxina

7.1.3 Meluxina-AI impact

In addition to the above considerations, LXP plans to acquire an AI infrastructure, Meluxina-AI. Meluxina-AI procurement is on-going, and the installation is expected later in 2026. The current Meluxina infrastructure will remain operational at least until 2027. However, for the commercial operation of the TEF, some adjustment might be required to migrate to Meluxina-AI. Work package 4 for example is already planning for compatibility with Meluxina-AI. The main difference between the current Meluxina (Cloud and HPC) and Meluxina-AI is the introduction of Kubernetes. SLURM (the current scheduler for Meluxina HPC) will still be available, however the Cloud partition will not. This migration can be performed later in the project, as only the Cloud infrastructure is impacted end of 2027.

To facilitate the migration to Meluxina-AI, the data services could be Kubernetes-based from the start. This Kubernetes environment would run on the same cloud VMs as specified above, with the option to offload some compute-intensive tasks to HPC nodes on demand. Unless the data services team has experience with Kubernetes, it is less risky to target a simpler environment and migrating it when required.

7.2 Conclusions

The HPC environment is positioned as a foundational enabler that supports the scalable execution of EnergyGuard services rather than as an isolated computational resource. The design recognises distinct roles for cloud-based virtual machines and HPC nodes, leveraging each where they are most effective in terms of persistence, connectivity, and computational performance.

This leads to a pragmatic hybrid deployment model in which long-running services such as databases, ETL orchestration, and semantic or query services are hosted on virtual machines, while compute-intensive or burst workloads are offloaded to HPC resources on demand. Storage and networking constraints are explicitly accounted for, guiding choices around shared file systems, data staging, and service placement. Together, these decisions ensure that the platform can scale computationally while remaining robust, maintainable, and aligned with HPC operational realities.

8 Data security and privacy mechanisms

This section describes the work performed in Task 3.6, that addresses one of the most critical dimensions of the EnergyGuard project: ensuring end-to-end security and data privacy across the entire Testing and Experimentation Facility (TEF).

The task focuses on designing and delivering a comprehensive security framework that safeguards data flows from ingestion to inference APIs, within both the AI development sandbox and the acceptance environments, as well as the end-user dashboard.

The work undertaken encompasses not only the technical implementation of robust anonymisation, encryption, and fine-grained access control mechanisms, but also the continuous reinforcement of legal, regulatory, and policy-based requirements.

By integrating advanced tools for identity management, penetration testing, and ethical hacking, the task ensures that the EnergyGuard platform is resilient against a broad spectrum of cybersecurity threats and remains compliant with EU protection regulations.

8.1 Security requirements

The following table lists the requirements concerning security aspects. The complete set of requirements is included in Deliverable D2.1.

Table 8: Security requirements

Req. ID	Functional / Non-functional	Related user story	Description	MVP Yes/No
UR-TEF-NF-002	Non-functional	US-TEF-018	Data should be anonymized and / or aggregated. All personal data shall be stored in encrypted form and managed in full alignment with GDPR.	Yes
UR-TEF-F-003	Functional	US-TEF-018	The system must allow secure access to all datasets and applications, with user authentication, permission control and data versioning. The system shall also request and record explicit consent from users before storing or processing any personal data.	Yes
UR-TEF-NF-003	Non-functional	US-TEF-003, US-TEF-004	The system shall ensure data traceability (access logs) and shall ensure data integrity through redundancy and recovery mechanisms.	Yes
UR-TEF-F-023	Functional	US-TEF-007	The system shall contain a modular testing suite based on standards such as ISO/IEC 27001 for security, and NIST AI RMF.	No
UR-TEF-	Non-functional	US-TEF-018	The system must adopt secure data sharing protocols and access control to protect IP and privacy.	Yes

NF-008				
UR-TEF-NF-009	Non-functional	Legal. Not derived from user story	The platform should be IP/Copyright compliant with regards to training, input and output data.	No
UR-TEF-NF-010	Non-functional	Legal. Not derived from user story	The platform should operate in compliance with AI Act if and to the extent applicable which needs to further be assessed. During the project, assessment for probable exceptions to AI Act's application with regards to AI systems developed, used or incorporated or any other operation will take place.	Yes
UR-TEF-NF-011	Non-functional	Legal. Not derived from user story	The Platform should be data protection/privacy compliant both at the development and operation phase (to the extent that processing of personal data takes place in each one of them) and will comply with data protection by design and data protection by default with regards to data lake creation, platform development, data processing, AI models, EnergyGuard website operation.	Yes
UR-TEF-NF-012	Non-functional	Legal. Not derived from user story	The platform should comply with the ethical requirements provided by "Ethics Guidelines for trustworthy AI" drafted by High level Expert Group on Artificial Intelligence (HLEG Guidelines) and by "Ethics by Design and Ethics of Use approaches for artificial intelligence" drafted by European Commission for the AI systems that will be developed and used to serve the purposes of the project.	No

8.2 Design of security components

8.2.1 User Authentication

8.2.1.1 Objectives and Requirements

User authentication is a cornerstone of the EnergyGuard "Foundational Framework." It is not merely a login function but the central service that establishes identity and, consequently, trust within the TEF ecosystem. The primary objective is to implement a centralized, secure, and scalable Identity and Access Management (IAM) service.

This service must decouple authentication logic from the individual services of the platform (e.g., Marketplace, Validation Platform, Data Repository), in line with the microservices architecture defined in D2.2. The design described here directly addresses the following requirements identified in D2.1:

- UR-TEF-F-003 (Secure access, user authentication, permission control): This is the primary functional requirement mandating a robust mechanism to verify who a user is and what they can do.
- UR-TEF-NF-003 (Data traceability, access logging): Strong authentication is the prerequisite for meaningful auditing. The identity established here (the "who") will be tied to every action recorded in the audit logs (described in section 8.2.3).

8.2.1.2 Tool Selection and Justification: Keycloak

As foreseen by Task 3.6 of the Grant Agreement [cite Grant Agreement], an evaluation of IAM tools was conducted. The selected solution for EnergyGuard is Keycloak. Keycloak is a mature open-source project (managed by Red Hat / IBM) specifically designed for identity and access management in modern architectures. It was chosen for the following strategic reasons:

1. Standards Adherence: It natively supports industry-standard protocols: OpenID Connect (OIDC) 1.0 and OAuth 2.0. OIDC will handle authentication (confirming the user's identity), while OAuth 2.0 will manage authorization (delegating access to resources/APIs).
2. Centralized Management (Single Sign-On - SSO): It provides a Single Source of Truth for identities. A user (e.g., "AI provider") will log in once to Keycloak and receive access to all EnergyGuard services for which they are authorized (the Marketplace, the testing portal, etc.) without needing to re-enter credentials.
3. Identity Brokering and Federation: In line with the D2.2 objective of aligning with the European ecosystem, Keycloak can act as a "broker." It can federate identities from external providers (e.g., GitHub, or academic/corporate IdPs via SAML), allowing users to use their existing accounts.
4. Security and Management: It offers a comprehensive administration interface for managing the user lifecycle (registration, password reset), configuring complex password policies, and enabling Multi-Factor Authentication (MFA).

8.2.1.3 Authentication Architecture and OIDC Flow

Keycloak will be deployed as a central, containerized service within the EnergyGuard architecture, acting as the sole Identity Provider (IdP). All other platform services (frontend and backend) will act as Relying Parties (RP) or Clients (OIDC terminology). The primary authentication flow will follow the OIDC "Authorization Code Flow" standard, which is considered the most secure for web applications: Accessing the Service (Client): A user attempts to access the "Validation Portal".

- i. Redirect to IdP: The Validation Portal, seeing the user has no active session, does not handle any credentials. Instead, it redirects the user's browser to the Keycloak login page. This request includes parameters like `client_id` (the portal's identifier), `redirect_uri` (where Keycloak should send the user back after login), and `scope` (e.g., "openid profile email").
- ii. User Authentication: The user enters their credentials (username/password) exclusively on the Keycloak domain. The platform services never see the user's password.
- iii. Authorization Code Issuance: After successful authentication, Keycloak redirects the user's browser back to the `redirect_uri` (step 2), including a temporary, one-time-use authorization code.
- iv. Token Exchange (Back-Channel): The backend of the Validation Portal receives this code. It then contacts Keycloak directly (over a secure back-channel, not visible to the browser) and exchanges the code (along with its own `client_secret`, a password for the service) for the actual tokens.
- v. Token Issuance: Keycloak validates the code and secret, and issues three artifacts:
 - a. ID Token (JWT): A signed JSON Web Token that proves the user's identity. It contains "claims" (information) like `sub` (Subject ID), `name`, `email`, and `iss` (Issuer, i.e., Keycloak). It is intended for the frontend to personalize the user experience.

- b. Access Token (JWT): A second JWT that represents authorization. This token is opaque to the frontend and is used to make secure API calls (e.g., uploading an AI model). It will contain claims essential for authorization, such as the user's roles (e.g., "ai-developer", "validator-admin").
- c. Refresh Token: A long-lived token used to obtain new Access Tokens (which are short-lived, e.g., 15 minutes) without forcing the user to log in again, ensuring a seamless but secure user session.

8.2.1.4 Identity Lifecycle Management and Security Policies

To complete the authentication framework, the following policies will be implemented via the Keycloak administration console:

- User Registration (Onboarding): A custom registration flow will be defined. New users (e.g., "SMEs") can register through a portal, but their accounts will be created in an "inactive" state. Mandatory email verification will be implemented; only after clicking the verification link will the account be activated and basic roles assigned.
- Multi-Factor Authentication (MFA): To strengthen security against credential theft, MFA will be enforced for all users with administrative roles. It will be offered for all other platform users, using standard methods like TOTP (Time-based One-Time Password).
- Password Policies: Strict complexity policies will be enforced (minimum length, use of special characters, numbers, upper/lower case), and checks against breached passwords will be enabled.
- Session Management: User sessions will be managed via token validity. Access Tokens will have a short lifespan (e.g., 15 minutes) to limit damage if stolen. Refresh Tokens will have a longer lifespan (e.g., 8 hours, or until inactivity) and will be centrally invalidated upon user logout.

8.2.2 User Authorization

Objectives and Access Control Strategy

While authentication verifies the user's identity, authorization determines access privileges to specific EnergyGuard platform resources (e.g., datasets, experiments, HPC computing resources, validation dashboards). The primary objective is to implement a Zero Trust security model, where no request is trusted by default, even if it originates from within the network. Every interaction with APIs or data must be explicitly authorized. The authorization strategy relies on two fundamental pillars to satisfy requirements UR-TEF-F-003 (Permission control) and UR-TEF-NF-008 (Secure data sharing protocols):

- 1) Role-Based Access Control (RBAC): For macroscopic permission management based on the user's job function (e.g., a "Data Provider" can upload data, a "Validator" can only read it).
- 2) Attribute-Based Access Control (ABAC) / Resource-Based Access Control: For more granular decisions based on resource ownership (e.g., an "AI Developer" can see only their own experiments, not those of other developers, despite having the same role).

Leveraging Keycloak's role management capabilities, the following standard user profiles will be defined within the EnergyGuard realm. These roles will be mapped into JWT tokens to be consumed by services.

- TEF Administrator: Has full access to all platform configuration features, user management, audit logs, and global resource configuration.

- Facility Operator / Infrastructure Manager: Responsible for managing physical and virtual resources (e.g., Digital Twins configuration, HPC monitoring). Has write permissions on infrastructural configurations but limited access to sensitive user data unless necessary.
- Data Provider: Users or entities (e.g., DSOs, building managers) providing raw data. They have permissions for:
 - Upload/Ingestion to the Data Lake.
 - Policy Definition on their own data usage (Data Sovereignty).
 - Visualization of usage statistics for their data.
- AI Developer / Service Provider: Users utilising the platform to develop and test solutions. They have permissions for:
 - Access to the AI Development Sandbox.
 - Read access to datasets for which they have obtained license/authorization.
 - Execution of training/inference jobs (with limited resource quotas).
- Validator / Auditor: Legal or ethical experts accessing the "Acceptance Environment." They have read-only permissions on experiment results, compliance logs, and access to robustness and bias analysis tools (e.g., ALTAI tool).
- End User / Consumer: Generic users who can access the Marketplace or public result dashboards (aggregated/anonymized data) for informational or service purchasing purposes.

Technical Architecture of Authorization

Authorization is not a monolithic block but is distributed across multiple layers of the EnergyGuard architecture to ensure Defense in Depth:

- I. API Gateway / Frontend Level (Enforcement Point):
 - i. The frontend uses information in the OIDC token (specifically the `realm_access.roles` claim) to show or hide UI elements (e.g., the "Delete Dataset" button is visible only to Admins).
 - ii. The API Gateway acts as the first line of defense, verifying the token signature validity and the presence of macroscopic roles required to access an endpoint (e.g., `/api/admin/*` requires the `tef-admin` role).
- II. Microservices Level (Resource Server):
 - i. Each microservice (e.g., Data Management Service, AI Orchestrator) implements fine-grained authorization logic. Upon receiving the JWT token, the service extracts the `user_id` (field `sub`) to verify resource ownership.
 - ii. *Example:* When an AI Developer requests `GET /experiments/{id}`, the service verifies not only that the user has the "AI Developer" role, but that experiment `{id}` actually belongs to that user (or is public).
- III. Data Storage Level (Data Lake & MinIO):
 - i. For direct file access (e.g., S3/MinIO object storage), Pre-Signed URLs or temporary tokens with limited scope will be generated. This prevents indiscriminate direct access to the Data Lake, forcing users to go through management services that enforce security policies.

Consent Management and Data Sovereignty

In line with requirement UR-TEF-F-003 (Request and record explicit consent), the authorization module will integrate mechanisms for consent management, which is particularly critical for personal or sensitive data (e.g., consumer data in Energy Communities).

- User-Managed Access (UMA) 2.0: The use of the UMA profile supported by Keycloak will be evaluated. This allows Data Providers to define granular policies (e.g., "Grant read access to dataset X to user Y for 30 days") without administrator intervention.
- Context Propagation: Information regarding consent and usage restrictions (e.g., "Research purposes only") will be associated with dataset metadata in the Data Catalog and verified at the time of access request (Query Engine).

Implementation Security Measures

- Short-Lived Tokens: Access Tokens will have a limited lifespan (e.g., 5-15 minutes) to minimize risk in case of token exfiltration.
- Authorization Auditing: Every authorization decision (granted or denied) generated by services or Keycloak will be recorded in the Auditing system to ensure full traceability.
- Environment Separation: Strict separation of permissions will be enforced between the development environment (Sandbox) and the production/validation environment (Acceptance), preventing a Developer from inadvertently altering validation data.

8.2.3 Auditing

The Auditing system is the component ensuring traceability of operations within the platform. Unlike technical debug logs (useful for developers to fix errors), audit logs have legal and compliance value, recording *who* did *what*, *when*, and with *what result*. This system is designed to meet the following critical requirements:

- UR-TEF-NF-003 (Data traceability, access logs): Obligation to maintain an immutable record of data access and critical operations.
- GDPR and AI Act Compliance: Necessity to demonstrate personal data security and transparency in AI model training/validation operations.

Not all operations will be logged to avoid log fatigue. The system will focus on security and critical business events, categorized as follows:

- Identity Events (IAM): Login (success/failure), logout, password change, user profile modification, role assignment.
- Data Access (Data Lake):
 - o Upload/Ingestion: Who uploaded a new dataset and when.
 - o Download/Access: Who downloaded or viewed a sensitive dataset.
 - o Modification/Deletion: Operations altering data state (e.g., soft-delete).
- AI Lifecycle (Sandbox & Acceptance):
 - o Start of a training experiment.
 - o Request for model validation.
 - o Issuance of a compliance certificate/trustmark.
- Infrastructure Management: Changes to Digital Twins or HPC resource configurations.

The auditing system will be centralized and decoupled from operational services to ensure log integrity.

1. Generation (Producers):
 - a. Keycloak: Natively generates login/logout and user management events. It will be configured to send these events to the central system.
 - b. EnergyGuard Microservices: Each service (e.g., Data Manager) will use a common client library to send audit events in a standard format (e.g., CloudEvents) upon executing critical actions.
 - c. API Gateway: Can log unauthorized access attempts (e.g., 401/403 requests).
2. Collection and Aggregation (Collector):
 - a. A centralized logging stack will be used (e.g., ELK Stack - Elasticsearch, Logstash, Kibana or Grafana Loki).
 - b. A message broker (e.g., Kafka or RabbitMQ) may act as a buffer to ensure no logs are lost even during traffic spikes.
3. Secure Archiving (Storage):
 - a. Audit logs will be stored in a dedicated index with WORM (Write Once, Read Many) policies to prevent modification or deletion, ensuring forensic evidence in case of incidents.
 - b. The retention policy will be configurable based on legal requirements (e.g., 1 year for standard logs, 5 years for access to financial/personal data).

To ensure interoperability and analysis, each audit record will follow a structured JSON schema containing at least:

- **timestamp:** Precise date and time (UTC) of the event.
- **principal:** Identifier of the user (User ID from Keycloak) or service initiating the action.
- **action:** Type of operation (e.g., DATASET_DOWNLOAD, LOGIN_FAILED).
- **resource:** The object of the action (e.g., Dataset ID, Experiment ID).
- **outcome:** Outcome of the operation (SUCCESS, DENIED, ERROR).
- **context:** Additional metadata (e.g., source IP address, User Agent).

8.2.4 Data Anonymization

8.2.4.1 Functional and architectural requirements

Data are considered anonymous if the identification of individuals, whether directly or indirectly, is not reasonably likely to occur, considering all means that can reasonably be used by any party to re-identify them.

The list of requirements for the TEF, included in section 8.1 includes one specific requirement for anonymization, that is requirement UR-TEF-NF-002: "Data should be anonymized and / or aggregated. All personal data shall be stored in encrypted form and managed in full alignment with GDPR".

In addition, deliverable D2.1 also contains one similar requirement for pilot 4, that concerns residential buildings. This is requirement UR-APP-NF-002: "Data should be anonymized and when not possible aggregated".

Deliverable D2.2, that describes the Energyguard architecture, specifies that data anonymization will be a standalone tool that will be used to anonymise data before their ingestion to EnergyGuard

platform” and that “Anonymisation applies data masking or pseudonymisation to protect personal/sensitive data”.

In the Grant Agreement, it is written that the Energyguard architecture will include “a data curation and anonymisation module responsible for ensuring that the real-world data are cleansed, curated and anonymised at the client/energy data provider side to safeguard the data privacy as well as to optimise the required resources”.

8.2.4.2 *Anonymization techniques*

There are several available techniques that can be used to implement data anonymization:

- Suppression¹⁵ consists in removing or hiding certain values that could identify a person. For example, replacing a person’s name or phone number with a blank.
- Generalization consists in making data less specific, e.g. by replacing an exact age with an age range, or a city with a region.
- Masking consists in replacing sensitive values with random or fake data that keeps the same format, e.g. a credit card number 4539-2456-9810-1234 replaced by “9999-0000-0000-1111
- Pseudonymization consists in replacing personal identifiers with artificial codes or “pseudonyms”. This is a reversible technique, since it is possible to keep a separate key that links pseudonyms to real identities. For example, user “Alice” could be replaced by “User_001”
- Aggregation consists in combining data from many individuals to show only group-level information. For example, instead of storing values for individual salaries, one could store the average salary per department.
- Noise Addition consists in adding random variations to numeric data so that exact values can’t be recovered. For example, one could add $\pm 5\%$ random noise to incomes or ages.
- Data Swapping consists in randomly swapping values between records to break the link between data attributes. For example, swapping postcodes between individuals.
- k-Anonymity and Related Models are formal privacy model that aim to quantify anonymity and reduce re-identification risk. These advanced anonymization techniques aim to ensure that everyone’s data cannot be distinguished from that of several others. They work by grouping records and controlling how unique or concentrated sensitive information can be within those groups, reducing the risk of re-identification or inference about any single person.

With *k-Anonymity*, each person’s record looks identical to at least k-1 others. With *ℓ-Diversity*, within each group of similar records, sensitive values are varied. With *t-Closeness*, the distribution of sensitive values in each group is close to that of the whole dataset.

Advanced anonymization tools include features to assess both residual re-identification risk and data utility loss.

Residual risk analysis estimates how likely it still is for an individual to be identified after anonymization, while utility analysis measures how much information value has been lost in the process. These two aspects must be balanced: stronger anonymization usually reduces data utility.

In cases where the residual re-identification risk remains high, encryption, described in next section, can provide an additional safeguard by protecting access to the data and ensuring that only authorized

¹⁵ <https://arx.deidentifier.org/>

users can view or process it. This combination helps strengthen overall data protection across both privacy and security dimensions.

8.2.4.3 *Tools for data anonymization*

For data anonymization, the following tools have been tested and evaluated:

- ARX Data Anonymization Tool¹⁶: this is a mature open-source tool, specifically designed for the anonymization of tabular data and with support for various privacy models (k-anonymity, ℓ -diversity, t-closeness, differential privacy) and residual risk analysis. It supports importing from CSV/Excel/DB, has a GUI interface and a Java library/API. It also offers data utility analysis (how much the transformation impacts the dataset) and management of generalization hierarchies.
- UTD Anonymization Toolbox¹⁷: this is an open-source research toolbox that implements some classic anonymization techniques (Datafly, Mondrian, Incognito with ℓ -diversity and t-closeness) for tabular data. It has good experimental foundation and may be useful for prototypes or research but is less "industrialized" than ARX.
- CloudTeams Anonymization Tool¹⁸ is a web-based data anonymization tool developed by the National Technical University of Athens (NTUA). It is implemented in Python using the Django framework, with a web interface built from HTML, JavaScript, and CSS. The tool enables users to upload datasets, configure anonymization parameters, and process the data through the web interface. It supports tabular data anonymization and includes modules for data masking, generalization, and suppression. It provides an interactive environment for inspecting datasets before and after anonymization.
- Amnesia¹⁹: this is an open-source tool for anonymizing tabular data, with a graphical interface, also designed for GDPR compliance. It provides a simple user interface, designed for less technical users, also aimed at ensuring data anonymization in compliance with regulations.

8.2.4.4 *Tool selection*

The following table shows a comparison between the tools mentioned in the previous section:

¹⁶ <https://arx.deidentifier.org/>

¹⁷ <https://labs.utdallas.edu/dspl/software/anonymization-toolbox/>

¹⁸ <https://github.com/eptu-ntua/anonymizer2>

¹⁹ <https://amnesia.openaire.eu/>

Table 9: Comparison between the tools for privacy and security

Tool	License	Privacy Models & Anonymization Techniques	Risk Analysis	Programming Stack	Integration API	Scalability	Community documentation
ARX Data Anonymization Tool	Open-source (Apache 2.0), developed by TU Dortmund	Supports k-anonymity, ℓ -diversity, t-closeness, δ -disclosure, β -likeness, δ -presence, and differential privacy. Transformation methods include generalization, suppression, micro-aggregation, sampling, top/bottom-coding.	Yes — provides re-identification risk analysis and data utility evaluation with visualization tools.	Java (desktop application + API)	Yes — Java API and CLI integration options.	Optimized for large datasets, scalable algorithms.	Active community; frequently used in academia and industry; regularly updated.
UTD Anonymization Toolbox	Academic project (University of Texas at Dallas), GPL license	Implements classic algorithms: Datafly, Mondrian, Incognito (+ ℓ -diversity, +t-closeness), Anatomy.	Basic privacy evaluation; minimal built-in utility analysis.	Java / MATLAB (some components)	Limited; primarily standalone use.	Works on moderate datasets	Development inactive since ~2012; academic use only.
Anonymizer2 (NTUA)	National Technical University of Athens; open-source	Masking, generalization, suppression.	No explicit risk or utility analysis modules.	Python (Django) web app	Web UI for real-time anonymisation	Intended for smaller datasets.	Limited maintenance and documentation.
Amnesia Anonymization Tool	OpenAIRE project (EU), open-source	Supports k-anonymity and k^m -anonymity via generalization and suppression; allows creation of hierarchies.	Visual feedback on data quality but limited quantitative metrics vs. ARX.	Java backend + Web UI (HTML/JS)	Web interface and RESTful API endpoints.	Works well for small/medium datasets	Used in EU OpenAIRE context; documentation available.

Based on the comparative analysis, ARX has been selected because it provides the most comprehensive and mature anonymization framework, supporting multiple privacy models, detailed residual risk and utility analysis, scalability, and stronger community support.

8.2.4.5 Data sources

The project integrates a wide range of data sources from the Testing and Experimentation Facilities (TEFs), including digital twins, microgrids, hydrogen platforms, and renewable-energy communities. Most of these sources provide technical, environmental, or infrastructure-level information and do not include personal data.

Only two groups of sources may contain information that is directly or indirectly linkable to individuals and therefore require particular attention regarding anonymisation:

- A. Residential Building Digital Twin Data: These datasets describe buildings, their energy performance, geographic attributes, and environmental conditions. While the data is primarily structural and operational, certain attributes—such as detailed location information—may act as quasi-identifiers when combined with external sources. As a result, appropriate anonymisation measures may be required before the data is used within the TEF environment.
- B. Renewable Energy Community Data: Data from producers, consumers, and prosumers include technical measurements from devices, operational time series, and energy-related metadata. Although mostly technical in nature, these datasets can contain some identifying or installation-specific information, which may need to be masked, generalised, or otherwise anonymised to safeguard privacy.

The project will adopt a selective anonymisation approach, applying privacy-preserving transformations only to those data sources that include personal or potentially identifying attributes, while all other TEF datasets can be processed directly without anonymisation obligations.

8.2.4.6 Anonymization process

For each dataset, an ARX project will be created. Dataset attributes will be classified according to the following categories:

- Identifying attributes, containing information that directly identifies a person. For example, name, social security number, telephone number, and email address. They must always be removed or completely masked.
- Quasi-identifying attributes, not identifying a person on their own, but when combined, making them identifiable. For example, date of birth, zip code, gender, and profession. They must be generalized or suppressed.
- Sensitive attributes, containing sensitive personal information that does not identify but should not be disclosed (e.g., health status, income, political orientation).
- Insensitive attributes, i.e. non-personal or harmless data that do not require any transformation.

For sensitive attributes, generalization hierarchies will be defined, to replace original values with more general or less specific ones to reduce the risk of identifying individuals. For example, age "27" could be generalized at level 1 into "20–29" and at level 2 into "< 40". Hierarchies allow anonymization algorithms to choose the right level of abstraction — balancing privacy protection and data utility.

After classifying attributes and defining generalization hierarchies, privacy models (such as k-anonymity, ℓ -diversity, or t-closeness) will be selected and configured according to the required protection level. ARX will then search for feasible transformations that satisfy the chosen privacy criteria while maximizing data utility. Finally, the resulting anonymized dataset will be evaluated using ARX's re-identification risk analysis and utility metrics, and the optimal solution will be exported for downstream processing.

8.2.5 Data Encryption

8.2.5.1 *Functional and architectural requirements*

Encryption techniques are applied to the database after data anonymization to ensure an additional layer of protection against unauthorized access and data breaches.

The list of requirements for the TEF, included in section 8.1 includes one specific requirement for encryption, that is requirement UR-TEF-NF-002: "Data should be anonymized and / or aggregated. All personal data shall be stored in encrypted form and managed in full alignment with GDPR".

In addition, deliverable D2.1 also contains one similar requirement for pilot 4, that concerns residential buildings. This is requirement UR-APP-NF-004: "All personal and building-specific data shall be stored in encrypted form and managed in full alignment with GDPR".

Deliverable D2.2, that describes the EnergyGuard architecture, specifies that encryption "protects data at rest and in transit"

In the Grant Agreement, it is written that the EnergyGuard architecture will provide encryption "across all the architectural components and the communication between them" and that "data encryption will be applied to protect security attacks and privacy breaches for all services".

8.2.5.2 *Encryption techniques*

Encryption techniques can be divided into two main categories: encryption in transit and encryption at rest.

Encryption in transit protects data while it is being transmitted between systems or over networks. Common techniques include Transport Layer Security (TLS) and Secure Shell (SSH), which use public key cryptography to ensure confidentiality, integrity, and authentication during data exchange. A widely used example of this is HTTPS (HTTP over TLS), which secures web traffic between clients and servers. At a broader network level, Virtual Private Networks (VPNs) also provide encryption in transit by creating secure, encrypted tunnels that protect all data flowing between a user's device and a remote network. Encryption at rest secures data that is stored on disks or within databases. This can be achieved using symmetric encryption algorithms such as AES (Advanced Encryption Standard), which encrypt and decrypt data using the same secret key. For databases, encryption can be applied at different levels — for example, full-disk encryption, file-level encryption, or column-level encryption, depending on security requirements and performance considerations.

8.2.5.3 *Adopted techniques*

For the implementation of these techniques, HTTPS and VPN are used together to protect data in transit.

HTTPS ensures end-to-end encryption between clients and servers, preventing interception or tampering with the transmitted data, while the VPN adds an additional layer of security at the network

level, encrypting all traffic between distributed system components and protecting against potential threats within intermediate networks.

The VPN will be used only when feasible, specifically to enable the different pilots to securely upload data into the data lake, whereas read-only access for other users will rely solely on HTTPS connections. For encryption at rest, a column-level encryption approach will be adopted to provide fine-grained protection of sensitive information while minimizing performance impact on database operations.

In PostgreSQL, this can be implemented using functions from the pgcrypto extension, such as `pgp_sym_encrypt()` to encrypt and `pgp_sym_decrypt()` to decrypt data at the column level, allowing secure storage of personal or confidential fields within otherwise unencrypted tables.

8.3 Towards implementation of security components

The implementation of the security and privacy mechanisms designed in Task 3.6 will be carried out within WP4 and WP5, across several tasks, and particularly in Task 5.5, "Security Testing, Vulnerability Assessment and Certification", led by the partner Hal Service with the contributions from other partners.

8.3.1 Data Anonymization

For what concerns anonymization, data will be anonymized before entering the platform at the providers side. The anonymization projects will be prepared in advance using example datasets. For each dataset type, the responsible partners will create a complete ARX project that already contains the attribute classifications, generalization hierarchies, and privacy model settings needed for anonymization.

To prepare the ARX projects, all dataset examples will be loaded into a staging area, checked, and their attributes will be classified as identifying, quasi-identifying, sensitive, or insensitive, and generalization hierarchies will be created for attributes that need them. The datasets will then be converted into clean, ARX-ready format.

The ARX projects will be configured by adding the dataset, attribute classifications, hierarchies, and privacy model parameters. ARX's optimization tools will be then used to automatically search for anonymization solutions that satisfy the selected privacy model while trying to preserve data utility. The best solutions, together with their risk and utility metrics, will be saved inside the project.

Once delivered, each data provider will simply open the project in ARX and replace the sample dataset with their own real data. This ensures that all sensitive or identifying information remains entirely outside the TEF environment and is never transferred inside.

The data provider can then run the anonymization using the preconfigured settings, review the suggested solutions, and export the anonymized dataset. This approach guarantees consistency across TEFs, while also ensuring that sensitive data never enter the TEF platform.

The anonymization process has the ability to evolve into a more automated and integrated service by relying on the ARX Java APIs to programmatically load source files, apply the required anonymization rules, and automatically deliver anonymized outputs to the EnergyGuard data lake.

Once the anonymization is completed, the service would export the transformed dataset in a standard schema and securely transmit it to the EnergyGuard data lake using encrypted channels. The raw data would never enter the TEF environment, ensuring full compliance with GDPR and project requirements while significantly reducing the risk of operator error and improving reproducibility across all TEFs.

8.3.2 Data Encryption

The implementation of encryption within the EnergyGuard platform will follow the principles and mechanisms described in Section 8.2.5, ensuring that all sensitive or personal data are protected both during transmission and once stored in the data lake. The objective is to guarantee confidentiality, integrity, and controlled access throughout the entire data lifecycle, in line with GDPR requirements and the project's security-by-design approach.

For data in transit, all communication between TEFs, ingestion components, and EnergyGuard services will rely on secure channels using HTTPS (TLS). Whenever required by the specific network configuration of a TEF, a VPN layer will provide an additional level of protection, ensuring that data exchanges take place within an isolated and encrypted communication tunnel. This combination prevents interception or tampering during dataset transfer and ensures that external actors cannot access the data flows.

For data at rest, encryption will be applied within the database layer after anonymization has been performed. To support fine-grained protection and minimise performance overhead, the platform will rely on column-level encryption techniques.

In the PostgreSQL environment used by the data lake, this can be implemented using the pgcrypto extension, which provides robust cryptographic functions such as `pgp_sym_encrypt()` for encryption and `pgp_sym_decrypt()` for controlled decryption. Only fields classified as personal or confidential will be encrypted, while non-sensitive fields will remain unencrypted to preserve query performance and analytical efficiency.

Access to encrypted fields will be strictly controlled through existing authentication and authorization mechanisms. Decryption operations will be permitted only to authorized services within the EnergyGuard backend and never to external users or TEF systems. Key management procedures will ensure that cryptographic keys are stored securely, are accessible only to designated components, and can be rotated or revoked in accordance with platform-wide security policies.

8.4 Data Protection Validation Framework and Assessment of Platform Functionalities

A Validation Process Framework to ensure a structured and systematic mechanism for securing that all data processing activities related to the functionalities of the Platform throughout the Project lifecycle are aligned with the applicable EU data protection regulatory architecture, including the GDPR, relevant EDPB Guidelines and CJEU case law, is established. The description of this Validation Process Framework will be included in the next version of the Data Management Plan. The Framework functions as a core compliance pathway, embedding data protection controls across the design, development, deployment, and ongoing evolution of the Platform's Features, while the ensuing privacy assessment whenever this is deemed necessary ensures that these controls are effectively and consistently implemented in practice. All involved partners have been informed of the assessment results from the validation activities conducted to date, as well as the corresponding corrective actions relevant to the features under their responsibility. All assessment outcomes are available in the internal, private project repository.

8.5 Conclusions

Data security and privacy are addressed through a comprehensive, end-to-end design that protects data throughout its lifecycle, from ingestion to access by AI services and user interfaces. The approach integrates technical safeguards with regulatory compliance, ensuring that personal and sensitive data are handled in accordance with GDPR principles such as data minimisation, anonymisation, controlled access, and auditability.

The architecture is anchored in centralised identity and access management with strong authentication, fine-grained authorisation, and a zero-trust mindset. Role-based and attribute-based access controls are combined to enforce both general permissions and ownership-sensitive restrictions across services and data layers. Privacy is operationalised by ensuring that anonymisation is performed outside the TEF environment while encryption in transit and at rest, together with auditing and consent-aware access, provides a robust foundation for secure and trustworthy data exchange across the EnergyGuard ecosystem.

9 Conclusion

The deliverable establishes the first integrated technical baseline of the EnergyGuard platform, defining how the project will support secure, interoperable and scalable testing and experimentation environments for AI-enabled energy solutions. It consolidates the core building blocks needed to collect heterogeneous data from multiple TEFs, harmonise it into a common structure, enrich it semantically, store it efficiently, and make it accessible for analytics, digital twins and AI workflows. As a whole, it provides a coherent “platform backbone” that enables later services and use cases to be implemented on a stable and consistent foundation.

A major outcome is the definition of a data-management approach centred on a modular, lifecycle-driven data lake. The architecture is designed to handle the full path from data acquisition to consumption, with explicit mechanisms for ingestion, harmonisation, storage, querying and controlled sharing. The document clarifies the need to accommodate both operational, high-frequency time-series data and more static datasets that provide context (assets, metadata, reference information), and it proposes a clear separation of these concerns while maintaining synchronisation between them. The updated architecture also incorporates object storage to support unstructured content and spatial capabilities (e.g., via a spatially enabled relational layer) to better reflect the realities of energy experimentation where files, documents, logs, and geo-referenced information often coexist with sensor streams.

The deliverable goes beyond conceptual architecture by describing concrete implementation work for ETL processes and the initial connections with real TEFs. It selects a specific ETL toolchain and explains how extraction, transformation and loading jobs are structured for maintainability and repeatability. It also adopts pragmatic engineering decisions to reduce integration complexity at this stage, such as using an intermediate transfer mechanism to simplify secure data movement from TEF sites into the platform. At the same time, the document is transparent about current maturity: only TEFs with already defined interfaces and accessible datasets are integrated in this first version, while other TEFs remain under definition and are planned for subsequent iterations. This staged integration strategy reduces risk and allows the platform to stabilise early with real data, while leaving room for progressive expansion.

Another important conclusion is the emphasis on harmonisation choices that directly support reuse and cross-TEF analytics. Normalising time-series outputs toward a consistent structure (timestamp, identifier, value and associated metadata) enables common storage, querying and downstream processing regardless of the original source. The use of a dimensional modelling strategy for time-series data is consistent with the need for efficient query performance and scalable reporting, and it reinforces the idea that experimentation should rely on dependable, well-structured data products rather than ad hoc extracts.

The deliverable also places strong focus on semantic interoperability as a prerequisite for meaningful experimentation across sites and domains. It frames the ontology and knowledge graph layer as the mechanism to unify concepts, relationships and metadata across TEFs, enabling richer queries, traceability and reasoning beyond simple time-series analysis. This semantic layer is positioned as the glue between raw measurements, contextual asset information and digital twin representations,

improving the ability to interpret results consistently, compare experiments, and automate parts of configuration and validation workflows.

Digital twins are presented as a central capability for EnergyGuard, not as isolated models but as platform-integrated services. The conclusions drawn from the framework suggest that effective experimentation requires digital twin models to be discoverable, configurable, and connected to the same harmonised data flows as the rest of the platform. By defining how digital twins are intended to interact with data storage, semantic models and orchestration services, the document lays the groundwork for later work on simulation-based validation, what-if analysis, and hybrid pipelines combining physical measurements with virtual model outputs.

Finally, the deliverable recognises that AI-enabled energy experimentation can be computationally intensive and must be supported by appropriate execution infrastructure. It therefore links the platform architecture with HPC-oriented capabilities, anticipating workflows that require scalable computation for training, validation and large-scale scenario execution. This is an important forward-looking element, as it acknowledges that data management and interoperability alone are not sufficient; the platform must also provide practical means to run demanding workloads reliably and securely.

In summary, D3.1 provides a solid first version of the EnergyGuard foundational framework by combining architecture, implementation choices and an incremental integration plan grounded in real TEF connections. Its main strengths are the end-to-end view of the data lifecycle, the practical ETL implementation approach, the explicit support for heterogeneous data types (time-series, static, unstructured and spatial), and the commitment to semantic interoperability as an enabling layer for cross-site experimentation. The staged nature of integration and the acknowledgement of TEFs still under definition indicate that the framework is intentionally evolutionary, with this first release establishing a robust base that can be extended and refined in subsequent deliverables as additional TEFs, services and use cases mature.

Annex I: References

- [1] EnergyGuard TEF partners, D2.2 - EnergyGuard Architecture in alignment with the European ecosystem, technical report, 31/08/2025.
 - [2] Hitachi, Pentaho Data Integration, <https://pentaho.com/products/pentaho-data-integration/>.
 - [3] J.L. Hernández, S. Martín, V. Lonac, I. de Miguel, A. Rossi, Data Quality in The Built Environment Across Europe, Proceedings of the 2024 European Conference on Computing in Construction.
 - [4] J.L. Hernández, C. de la Cruz, I. de Miguel, F. Vélez, Enhancing Data Quality via Data Correction Techniques to Assure High-quality Data for Energy Services Across Europe, Proceedings of the 2025 European Conference on Computing in Construction.
 - [5] Microsoft, Star schema, <https://learn.microsoft.com/en-us/power-bi/guidance/star-schema>.
 - [6] Towards data science, Star schema, <https://towardsdatascience.com/tagged/star-schema>.
 - [7] <https://www.nrgexpert.com/energy-market-research/electricity-transmission-and-distribution-report-and-database/>
 - [8] <https://datacatalog.worldbank.org/search/dataset/0038055/Derived-map-of-global-electricity-transmission-and-distribution-lines>
 - [9] https://www.academia.edu/108514176/Big_data_analytics_in_power_distribution_systems
 - [10] <https://www.nitorinfotech.com/blog/apache-trino-introduction-to-the-powerful-query-engine/>
-