



Project co-financed by the European  
Regional Development Fund

# **D3.4.1**

## **WP3 Testing – WG2**

### **Operational Database**

## **Final Version**

### **SHERPA**

SHared knowledge for Energy Renovation in buildings by Public  
Administrations.

Priority axis-Investment Priority-Specific Objective 2-1-1

#### **Partner in charge (Authors)**

Stoyan Danov (CIMNE UPC),

Josep Mayos & Xavier Cubillas & Jaume Palmer (CIMNE UPC)

Supported by LP-DTES – Gencat

Xavier Martí & Silvia Mata

Final version prepared by LP DTES for A.3.4 – Information System  
Distribution: Public  
Date: August 30<sup>th</sup>, 2018

## Content

<b>1. Introduction</b>	3
<b>2. Development of the work</b>	4
<b>3. The SHERPA Operational Database</b>	5
3.1. Requirements	5
3.2. System architecture	7
3.2.1. SHERPA Application	8
3.2.2. Big Data Engine	9
3.3. Description of the Database Structure	10
3.3.1. Data groups in SHERPA Database	10
3.3.2. Energy consumption data allocation within the system	12
3.3.3. Data cleaning procedure	12
3.4. Benchmarking and baseline calculation modules	13
3.4.1. Benchmarking module	13
3.4.2. Required inputs	14
3.4.3. Consumption prediction module	16
3.5. Functionalities of the SHERPA Application	18
<b>4. References</b>	24
<b>5. Annexes</b>	25
5.1. Description of the SHERPA API	25
5.1.1. SHERPA Application API	25
5.1.2. Big Data API	25
5.1.3. REST methods	26
5.1.4. Input Resources	26
5.1.5. Output Resources	27



## 1. Introduction

The operational database for the analytics developed in the project has been implemented as a software resource within the SHERPA platform and is a fundamental part of the whole information system. The SHERPA platform has been developed by CIMNE on the base of its existing Big Data Analytics architecture, taking into account the requirements of the participating public administrations. The purpose of this document is to provide a general description of the platform and operational database, with an introduction to the big data processing architecture including both the back end and the front-end sides. A demonstration of the functionalities is also shown through concrete examples of the main features of the platform.

In order to provide a more global view of the SHERPA system, necessary to understand the database's development in the specific context of the project, the collected requirements of functionality, organization and access to the information by the users are described. These requirements have served as a general guidance for development of the SHERPA system and its components. In continuation, the global architecture of the SHERPA system is presented and the location of the operational database in the set, as well as the relationship with the other elements of the system is explained.

In section 3.3 a description of the database structure is presented and the main groups of data that have been implemented to ensure the functionality of SHERPA are indicated. Due to the relevance of the data provided by the users, which together with the consumption data and meteorological data represent the main basis of the analytics in the advanced modules, these are presented in more detail. The different groups of user data are then described more in details, together with the API (Application Program Interface) used for the data uploading process.

Finally, the benchmarking and consumption prediction modules are described, while in section 3.5, concrete examples of the SHERPA Application's functionalities are presented and related with the user requirements they are fulfilling.



## 2. Development of the work

The SHERPA Information System is based on the Big Data Architecture of CIMNE, which has been complemented and adapted to the requirements of the SHERPA project. The software development, including the creation of the SHERPA's Operational Database, the analytics and the user interfaces have been done by CIMNE, with the support of subcontracted organisations which contributed to specific tasks. The work includes broadly the following:

- Development of the SHERPA's Application Graphic User Interface (GUI)
- Development of the back-end for user management
- Development of the APIs that enable the communication between the SHERPA's web Application, the Big Data platform and the external service providers
- Development of new analytical modules allowing the upload of weather data from external meteorological stations and the calculation of baseline consumption for evaluation of energy savings, benchmarking and recommendations
- Development of technical documentation and user manuals



### 3. The SHERPA Operational Database

#### 3.1. Requirements

The preliminary system functional requirements have been collected from the users participating in the SHERPA pilot sites. Based on a pre-defined set of functionalities defined by CIMNE and DTES and used as a precursor, the users have been asked about their preferences, needs and expectations for the services to be finally delivered by the Information System. The collected requirements have been categorized according targeted user interactions and agreed upon with the representatives in each participating public administration. These requirements, adapted and further detailed during the testing phase, have been guiding the technical development work in the project.

The main functional requirements defined for the Information System are summarised below:

- Track energy performance in detail
- Communicate energy performance in a user-friendly manner
- Benchmark building energy use over time and to similar buildings
- Collect applied Energy Efficient Measures (EEM) and the related investments
- Evaluate the effectiveness of the applied EEM, learn and share the knowledge among the users
- Facilitate the management of energy efficiency intervention plans by collecting all the necessary information in a single place
- Enable a possibility to export results

In addition to the above functionalities, some specific characteristics of the system have been required:

- Necessity of user authentication and traceability of access within the system
- Different levels of user credentials within the organisation (administrator, user)
- Possibility for automatic and manual energy data upload

The rationale of each user requirement is extended below by including more details about the intended functionality and type of users supported.

##### Track energy performance in detail

Tracking energy performance requires a baseline consumption model for each dataset. Using a baseline model, it is possible to generate a counterfactual forecast of the building energy consumption, as if no changes had occurred since the baseline period. A comparison between this forecast and the actual consumption will enable users to track changes in the building performance over time.

There are two possibilities to forecast building performance based on energy consumption from previous periods, referred to as a baseline:

- using a fixed baseline period (fixed reference period in the past)
- rolling baseline (period with fixed length moving ahead with time, normally the last 1 or 2



years before the actual moment).

A model based on a fixed baseline period allows for cumulative savings to be tracked, and is usually used for assessment of action plans with a predefined duration. Using a rolling baseline allows for a performance metric that adapts to the changing building behaviour which encourages continuous improvement. Both of the approaches are useful for the public administration implementing SEAPs (Sustainable Energy Action Plans) and building renovation.

### **Communicate energy performance in a user-friendly manner**

Energy performance calculations are complex and sometimes difficult to present in a way easy to understand. The detailed visualisations necessary to track performance contain large amounts of information and require a certain level of knowledge, and take time and attention even for experts to interpret. Producing simple summary information in an easy-to-absorb format helps experts to understand where they should 'drill down' and look at the detail. Since many users will not be experts in energy in buildings they will benefit from a summary report expressed in a simple but engaging form and may not want more detailed information. For those who want more information, an extra click through will provide the detailed analysis.

For instance, presenting the actual consumption compared with the baseline and with quantification of the deviation between them in percentile scores provides a concrete and objective measure of 'low', 'medium' and 'high' consumption.

These values can be equated with 'good', 'neutral' and 'bad' energy performance, respectively by visualising them as happy, neutral or sad smiley faces.

### **Benchmark building energy use over time and to similar buildings**

The benchmarking should allow energy performance comparison between buildings of the same type (and subtypes) and allow the detection of buildings with greater potential for savings and prioritization of intervention. It should also be possible to identify the buildings with the highest performance, in order to allow detection of good practices to increase the energy efficiency.

The benchmarking should also allow to compare the actual energy performance of the building with its performance in previous periods.

### **Collect applied EEM and related investments**

Public buildings undergo different energy efficiency interventions over the time, and these interventions are related with considerable investments. There is a necessity to collect all this information in one place in order to be able to analyse it and adjust the future action plans according the obtained results.

A summary of past EEM in the building should be available together with the corresponding investments done in order to facilitate the assessment of their effectiveness and decision making for new measures to be applied in the future.



### **Facilitate the peer learning and manage an energy efficiency intervention plan**

Collecting the detail of different energy efficiency interventions and experiences from actions carried out in buildings with similar typology in other organisations can serve the users to assess in advance the potential investment and effect from applying them to their own buildings. Benchmarking the energy consumption of buildings with that of other similar buildings can reveal the available energy saving potential and help planning future actions.

The SHERPA Information System should combine both of the above functionalities in a single user interface in order to allow the public administrations to identify the saving potential of their buildings and improve their energy efficiency intervention planning based on the previous experience of their peers.

### **Authentication and user traceability**

All the target users should have access to the SHERPA system with their personal identification. Data is to be protected from protection against unauthorised users.

### **Different levels of user credentials within the organisation**

At least two different levels of user credentials should be allowed in the platform, *administrators* having permission to create, delete, edit and locate buildings and groups of buildings, and *users*, able to access and export the results of the analytics, but not to edit uploaded information in any way.

### **Possibility for automatic and manual energy data upload**

The system should allow both automatic and manual energy data upload. Automatic upload should be possible to configure in case disposes of energy management or energy accounting systems in order to facilitate the data transfer. An option for manual upload of data should exist for users not disposing of such energy management systems.

## **3.2. System architecture**

The SHERPA Information System is intended to be a repository where to store the energy consumption and the energy efficiency measures implemented in different buildings of an organization. Its objective is to facilitate the monitoring of the whole building stock's performance in a single place and provide useful output for different types of users within the public administrations, including energy managers, policy and decision makers, financial officers.

The system architecture is based on the Apache Hadoop open source software framework for Big Data storage and analytics. This makes the system easily scalable to large number of public administrations and hundreds of thousands of buildings.

The core of SHERPA is the data storage and analysis of energy consumption data adapted to different energy consumption data granularities, with total flexibility. The energy consumption is injected and stored into the system paired with a timestamp and not in fixed intervals. This allows the system to operate with either monthly data from energy bills, or hourly or sub-hourly data from smart meters,





and to produce analysis and results to the extent permitted by the available data.

SHERPA’s data can be introduced manually or automatically by exchanging data with existing building energy management systems (BEMS).

The architecture of the application is based on two big parts. The first part is the Big Data Analytics Engine, which contains the Common Operational Database and the data analytics software modules. The second part is the SHERPA web Application (referred in the text also as SHERPA App), which is the user interface for interacting with the database and administrating the users of the system.

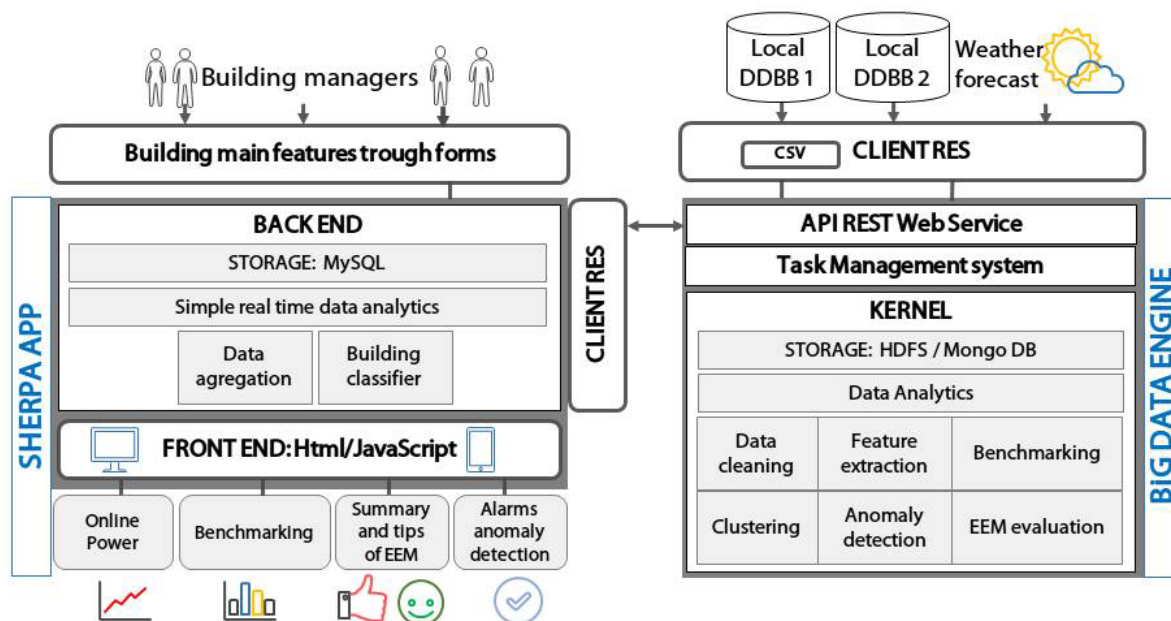


Figure 1 - The SHERPA architecture

### 3.2.1. SHERPA Application

The main function of the SHERPA Application is to allow the interaction between the end-users (building managers) and the Big Data Engine. It facilitates the settings editing, the data importing and the visualization of results. It is programmed under the Django framework. Django is a collection of Python libs allowing users to quickly and efficiently create a quality Web application. The SHERPA Application has a *Front End* for user interaction and a *Back End* for interaction with the Big Data Analytics Engine

**Back End.** The back end comes with an Object Relational Mapping that allows to manipulate data sources with ease. It provides an abstraction layer, the models for structuring and manipulating the data and facilitates the creation of forms to process user input and validate data and signals.

The back end is made up of two components: the *storage* and the *real time data analytics*. The storage is a structured open source database, based on MySQL. It is linked to the REST (Representational State Transfer) client communication layer and stores the information





introduced by the building managers through the web forms and the treated information coming from the big data engine. The real time data analytics is formed by several Django query sets which execute data manipulations on the database. Data aggregation, in several time frequencies, as well as a building classifier based on the categories defined by the building managers, constitute the main data operations performed by the Back End.

**Front End.** The front end helps with data selection, formatting and display. It features URL management, a language template, authentication mechanism, cache hooks and various navigation tools such as paginators. The front end support JavaScript (JS) programming language and shows, interactively, the data streaming and the results of the Data Analytics modules.

### 3.2.2. Big Data Engine

The Big Data Engine is designed to tackle the following Information Technology (IT) challenges:

- to offer on demand high quality analytic services
- to ensure big and scalable data storage and processing capacity
- to provide batch-processing data analytic services
- to ensure data privacy and security

**API REST.** This is the communication interface between the REST Client of the SHERPA App and the other local databases which provide the smart meter readings of the buildings. The API is fully developed following the REST programming style. The aim is to enable a Service Oriented Architecture (SOA), offering specialised energy services to the building managers. The main functions of the API are to import data into the Engine and to export data from the Engine. These objectives are addressed using different technologies. Data import and export are enabled using the Eve framework to implement the Web service. OpenAM provides open source Authentication, Authorization, Entitlement and Federation software. The Flask and Python modules implement all the server functionalities in order to deploy a web API server.

**Task Management Service.** This level is in charge of scheduling and synchronizing the tasks in the engine by means of RabbitMQ and Celery. The scheduler picks up the new task to be executed into the Engine according to a scheduling policy. The Quality of Service (QoS) is based on task return time and not on response time. Therefore, when a task enters the engine, it continues its execution in a batch mode until finalisation. Celery is the scheduler itself. RabbitMQ is a fast internal message-queuing system used to interchange information between tasks with different paradigm technologies.

**Kernel.** The Kernel (core part of the software) is made up of two main components: *Data Analytics* and *Storage*. Data Analytics is a comprehensive set of modules that enable the simultaneous parallel processing of big quantities of data in order to generate the required results in a reasonable time, in other words, it is the set of modules in charge of processing the stored data. It is a combination of R and Python software libraries to allow complex calculations and data-mining tools in a Big-data environment. The combination of both packages permits the calculations to be optimized and the data processing time, reduced.



Storage is the system that allows the large amount of data produced to be stored and managed in a scalable way. It is made up of a combination of low-cost hardware and database technologies that allows the acquisition, allocation and extraction of big quantities of data in a scalable manner to be processed by the analytics module. The short-term database uses MongoDB and the long-term database uses a Hadoop distributed storage technology. The short-term database is used as a buffer storage for data reception and sending in fast environments. It is the first data storage directly connected with the API and provides high communication bandwidth and scalability performance. It supplies temporary storage, acting as a cache memory, prior to it being permanently stored in the long-term database. HBASE is the long-term Big-data storage where all the historical data is stored and the analytical modules operate. The main benefit of using the Hadoop technology is that it allows the process and analysis of large volumes of unstructured and semi-structured data in a cost- and time-effective way.

### **3.3. Description of the Database Structure**

The SHERPA analytical databases are strategically hosted within the Big Data platform in a way that allows ensuring the system's functionalities such as:

- Store consumption data and buildings and relate them to each other
- Provide input for the analytical modules
- Store the results of calculations
- Feed applications and the front-end with data
- Extend seamlessly the storage capacity in case of necessity

The databases are divided into two parts within the system:

- Intermediate repository (buffer) in MongoDB that is used to host the data received in the system and the calculated results needed to feed the components that make up SHERPA and the front-end. This repository stores the data only for the time necessary to ensure the smooth operation of the applications and the visualization in the front-end.
- HBASE long-term repository in Hadoop that is used to host all the raw data, transformed data, and calculation results generated during the service needed to feed analytics, benchmarking and back-up storage. Due to the distributed nature of Hadoop this storage has enormous capacity. By distributing storage and analytics between different machines, Hadoop also divides the workload and can execute multiple tasks in parallel.

#### **3.3.1. Data groups in SHERPA Database**

Below are the data groups that make up the SHERPA analytical databases. These groups follow the organization of the data in MongoDB but have direct correspondence in HBASE, where they are transferred through ETL functions (Extract, Transform, Load), which at the same time transform the data into uniform physical units to prepare them for their correct analytical treatment.



**Table 1 Data groups for analytical databases**

Data groups in the SHERPA Databases	Description
/amon_measures	Contains all the hourly and sub-hourly consumption data uploaded in the system
/tertiary_amon_measures	Contains all the consumption data coming from bills (non-hourly frequency)
/baselines	Contains all the consumption baselines associated with each building that are used to evaluate savings or for the classifications that are part of benchmarking tool
/buildings	Contains all the available information about the buildings (year of construction, area, typology, ...)
/similar_groups_best_criteria	Group of support data for benchmarking: contains comparison groups resulting from clustering by similarity
/similar_groups_description	Group of support data for benchmarking: contains the detailed description of the parameters of each group resulting from the clustering
/league_table	It contains the classifications of all buildings according to their consumption evolution (increase / decrease)
/modelling_units	Contains the description of all the energy meters associated with each building and all the combinations of these meters (aggregations of measurements) defined by the SHERPA applications that need analysis and visualization in the front-end
/raw_data	Contains the raw data, just as they have been entered into the system. The consumption data is saved with its original granularity (timestamp / value)
/postal_codes	Contains all the postal codes of the European countries (used to link the location of the buildings with the weather stations)
/stations_measurements	Contains the data of the meteorological stations (ID, GPS coordinates, height, parameters measured in the station in timestamp / value format)



### **3.3.2. Energy consumption data allocation within the system**

The SHERPA system is flexible with respect to the energy consumption data granularity and allows operating with data coming from energy bills or from smart meters. The energy billing periods are usually monthly or bi-monthly and may not necessarily correspond to the natural months (e.g. from day 1 to day 31).

SHERPA stores the introduced energy consumption data into the system as pairs of (timestamp [UTC date-time], energy consumption [kWh]) and keeps both the original data set (raw data) and a transformed set of data in natural months. The raw data set is kept for traceability of the user's transactions and for security. However, for the analyses the energy consumption in hourly and natural month granularity is used. This permits to produce consistent results and visualisation of the energy consumption in the user interface.

The energy consumption data transformation and allocation for analysis within the SHERPA system is done in the following way, depending on the original data granularity.

- Monthly or larger period data from energy bills:

The energy consumption is distributed evenly in daily consumption data for the period and then integrated in natural months.

- Hourly or sub-hourly data:

The energy consumption is integrated first in daily consumption and then in natural months.

- When monthly and hourly data are available:

The monthly data are distributed evenly in daily consumption data for the period. The hourly data are integrated in daily data. Both sets of daily data are aligned and overlapped. Where hourly data exist, these are given priority and overwrite the monthly data in the final set. This procedure is intended to cover cases where there might be some gaps in the hourly data in order to ensure consistent set of data for the analyses.

### **3.3.3. Data cleaning procedure**

The quality of the data inserted into the SHERPA database is critical for the quality of the services. Due to the possible big differences of the energy consumption between different buildings, and even between the consumption in different months for the same building, it is not possible to apply any meaningful procedure for detecting wrong data at the moment of their insertion. The SHERPA system relies that in principle the inserted data by the user is correct. Incorrect data affects the analyses and results produced by the system and it is responsibility of the user to detect and correct any possible errors in the data that he or his data provider inserted.

During the period of analysis after the data insertion some automatic error detection and data cleaning is possible for hourly or sub-hourly data acquired by automated measuring meters. However, automated data cleaning is not possible for monthly data and these should be corrected by



the user himself by re-inserting them.

The data cleaning process for hourly and sub-hourly data implemented in SHERPA is explained in continuation.

The data cleaning is a critical process for the correct analysis of the data and is implemented over the raw data set. The data cleaning is basically focused in the detection of “outliers” and “gaps” within the data consumption series. Different procedures for data cleaning exist, and an extended revision of the available methods is presented in (Acouemo 2015). In the case of SHERPA two methods are applied:

- i) Non-recursive elimination of extreme points based on Z-scores in a temporal window of one week ( $Z_t = (y_t - \mu)/\sigma$ ), where  $y_t$  is the energy consumption in the instant  $t$ ,  $\mu$  is the centred average of the hourly values during the week and  $\sigma$  is the standard deviation. The energy consumption value is considered an outlier if the  $Z_t > 3$ .
- ii) The second method is based on the theoretically evaluated maximum consumption calculated from the contracted power for the building. If the energy consumption overcomes this maximum theoretical value it is considered an outlier. The days containing outliers are eliminated from the dataset and are not considered in the prediction models.

For the detection of data gaps the “data padding” algorithm which considers the original frequency of the data series is used. The timestamps of the erroneous data are substituted with NA (not available). All the process of data cleaning is executed in a parallel manner in order to reduce the computational time.

### **3.4. Benchmarking and baseline calculation modules**

#### **3.4.1. Benchmarking module**

This section describes the benchmarking tool implemented in the SHERPA system that allows users to access comparisons and detailed metrics about their building energy performance.

The benchmarking module produces comparative analysis of the energy performance for each specific building over the time, or compared with buildings of the same type (and subtype). The analysis serves for detection of the savings’ potential and prioritization of energy efficiency interventions. The benchmarking involves multiple separate analyses for each individual building which are supported by the powerful Big Data analytics and parallel processing technologies in the core of SHERPA platform. Buildings with better energy performance are identified and their good energy efficient practices are detected from the information in the database and used for peer learning.

The energy performance of the buildings is evaluated through key performance indicators (KPI) which are calculated and compared to those of other buildings clustered in multiple groups according to different similarity criteria (see next section below). The consumption indicator per unit area



(kWh/m<sup>2</sup>) on annual basis is considered as a starting point, and may be completed with other common indicators (e.g. kWh/user) or specific to a given building typology (e.g. kWh/students of an educational centre).

The benchmarking results are offered to the user allowing him to assess the building's performance from multiple perspectives; for example, how a school building performs compared to school buildings in the same geographic region, or to other school buildings of the same public administration in different geographical regions, or to school buildings of other public administrations. The SHERPA's comparative analysis enables assessing the energy saving potential and prioritising of interventions and at the same time offers recommendations to the users based on the best performing schools' practices.

### 3.4.2. Required inputs

For the development of benchmarking the following data are required as inputs:

- Use typology and sub-typology (e.g. typology 'police station', sub-typology 'large police station').
- Values used for the calculation of the ratios or indicators (KPIs) chosen for the benchmarking (e.g. surface, parameter for the calculation of consumption indicator kWh/year/m<sup>2</sup>).
- Energy consumption
- Postal code, used to automatically link the building with the location and the reference Weather Station for climate normalization

#### **Similarity criteria and parameters**

One of the goals of the SHERPA benchmarking module is to give a quick idea of the building's energy performance compared to similar buildings. Since the similarity can refer to different aspects relevant to the energy performance of buildings, the tool allows a custom user configuration. Some similarity parameters are defined (Table 2), depending on the data availability, for each building they can be combined to obtain different similarity criteria (Table 3).

**Table 2. Similarity parameters for grouping buildings for benchmarking**

Similarity parameter	Description
<b>entityID</b>	Name of the organization
<b>useType</b>	Building use / Building typology
<b>constructionType</b>	Building type (structural)
<b>occupancyHours</b>	Hours of occupancy (opening and closing time of the building)
<b>dayWeek</b>	Number of days the building is open during the week
<b>postalCode</b>	Postal code

The similarity criteria are defined based on the similarity parameters. The more parameters are included, the more restrictive the criterion is, since all the included parameters must be matching for a comparison group of buildings. The Benchmarking module of SHERPA offers comparatives among





multiple groups of building created on basis of the similarity criteria for benchmarking in Table 3. The comparatives are launched for execution simultaneously for all of the criteria and results are produced for all of those the available data permits.

The procedure operates like this: The system evaluates the results for each of the imposed criteria by studying the number and the dispersion of the samples. Criteria resulting in selection of less than 5 buildings are omitted and no results are produced for them, a threshold condition set to ensure representativeness and quality of the results. For the criteria that overcome the threshold results are calculated and offered to the user. The system ranks as the best the similarity criteria producing enough samples that really present a similarity in consumption, that is, those having the most compact distribution of the energy consumption values.

The SHERPA App visualizes by default the benchmarking with the criteria that best represents the similarity in consumption. Nevertheless, all the rest of comparatives are also available and the user can check them through an option in the user interface.

**Table 3 Similarity criteria for benchmarking**

Similarity criteria for benchmarking
<i>entityId + useType + constructionType+ occupancyHours+ dayWeek</i>
<i>entityId + useType + constructionType+ dayWeek</i>
<i>entityId + useType + occupancyHours+ dayWeek</i>
<i>entityId + useType + dayWeek</i>
<i>entityId + useType</i>
<i>entityId + constructionType+ occupancyHours+ dayWeek</i>
<i>entityId + constructionType+ dayWeek</i>
<i>entityId + constructionType</i>
<i>entityId</i>
<i>entityId + postalCode + useType</i>
<i>postalCode + useType</i>
<i>useType</i>

For example, if the considered building has no data entered except from the energy consumption, it can only be compared with the ones of the same organization (entityId). If more parameters are present in the database (useType, constructionType, occupancyHours, daysWeek), the system compares and selects in which criteria there are enough samples for the analysis and from them those that show a better similarity of consumption.





### 3.4.3. Consumption prediction module

#### Scope and definition

This module consists of the implementation of a monthly or hourly regressive consumption prediction model, depending on the level of granularity of the available data. Both models are based on the historical series of energy consumption of the building from at least one previous year. The monthly model applies climatic normalization of the consumption, while the hourly model applies additionally normalization for the degree of building occupation hours in case the data is available.

The monthly prediction model is the model applied by default when hourly data is not available. The hourly model allows for better prediction and with the recent availability of real hourly consumption curves for some buildings provided by the electricity supply companies, and the incorporation of telemetering, can be applied to an increasingly larger number of buildings.

#### Required inputs

The following information is required:

- Monthly or hourly consumption data
- Data for climate normalization
- Data on building occupation hours

#### Technical implementation: baseline

The visualization system is based on historical data comparisons. To generate a robust comparison of historical data it is necessary to build a model that captures and takes into account the variation of consumption due to occupancy and differences in outside temperature. The model for calculating the baseline can be constructed with monthly consumption data or with hourly data. Both models generate a 12-month baseline consumption that is used to assess the evolution of the building performance over time.

In case of having hourly data, the model of the baseline allows taking into account both the variation of the temperature and the variation of occupation with respect to the current period and introduces the necessary corrections for a good evaluation of consumption, according to the recommended procedure of IPMVP (International Performance Measurement & Verification Protocol).

With the hourly consumption for each week (Monday to Sunday), the baseline period is formed from the 12 months preceding the current week. Using this kind of "rolling" baseline that gets updated as time passes, allows the model to adapt to the "normal" consumption of the building and reflects only the persistent changes in its behaviour.

The occupation profiles of non-residential buildings (offices, public buildings) are quite repetitive and follow a weekly pattern. The peaks of occupation during weekdays are reduced during the night and on weekends. Both electricity and gas consumption are affected by the day and time of the week, which can be used as an indirect approximation for the occupation. The effect of the time of the week can be captured (in the case of hourly data) by separating the base period data set into 168



corresponding subsets of each hour of the week and finding a separate model for each subset.

For example, the procedure would be to find and apply 168 models, one for each hour of the week, using the approximately 52 data available in each of the subsets of data. The simplest possible model would be the average consumption of each hour of the week. However, when the energy in the building is consumed for heating or cooling purposes, the variation of the outside temperature cannot be ignored. In this case, a consumption model of three parameters is derived for each set, using the external hourly temperatures. In this case, the baseline model is dependent on occupation, as well as outside temperature.

In many cases, especially when the energy consumption of the building is not heating or cooling, it is expected that the consumption does not depend on the outside temperature. In these cases the Bayesian Information Criterion (Stuart 2011) is used to determine in which time period to apply the most complex model and in which to apply the simple model of average consumption. This allows avoiding the use of over-defined models.

To extend this model (or any model) to a "normal range", the residuals (measured as the difference in consumption and prediction) are calculated for each data point. For each of the 168 subsets the values of the 10, 25, 75 and 90 percentiles are determined. These values represent the variation from the model in each hour of the week. These values are used to visualize the range of expected consumption as 3 zones of probabilistic expectations around the model. If consumption is in the area of consumption corresponding to the 0-25 percentiles, it is assumed that it is in the good zone. Between percentiles 25-75 consumption is considered in the neutral zone, and above the 75 percentile is considered high consumption, see figure below.

The model is derived from the raw consumption data (to the right) and the model residuals (to the left) are used to generate the coloured zones that are then applied around the average consumption to represent the expected consumption for each temperature.

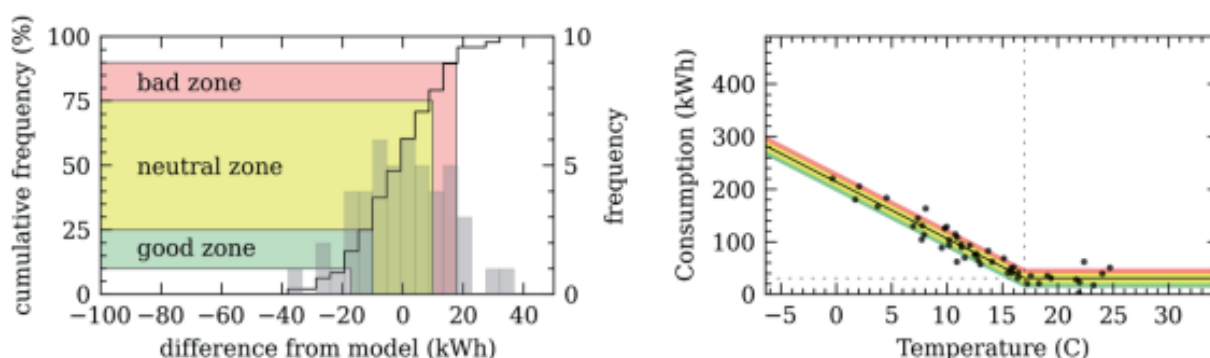


Figure 2 Model derivation and generation of coloured zones



### 3.5. Functionalities of the SHERPA Application

Following, the different features of the application are briefly presented with some screen-shots of the user interface and a reference to the different functional requirements mentioned at the beginning of the document.

#### Track energy performance in detail

From the building dashboard it's possible to access a detailed description of the building's energy performance in terms of consumed kWh (total and per square meter), saved energy kWh, and an energy performance indicator obtained through a comparison with the other buildings in the database.

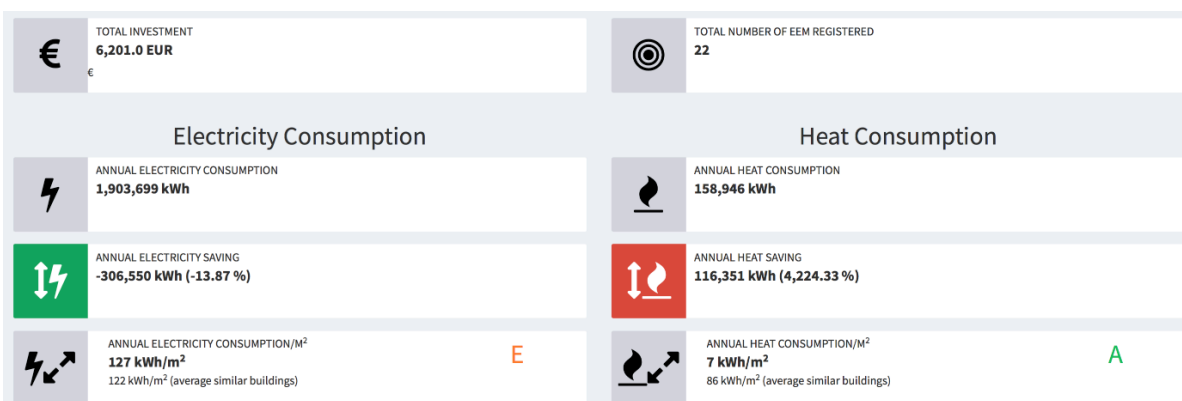


Figure 3 SHERPA building dashboard

Graphs showing the evolution of the electricity and heat consumption during the last year are also available, in the same page. The climatically adjusted baseline consumption is shown on the same graph which permits to assess deviations and follow-up improvements in the building performance. Both electricity and heating energy consumption can be displayed.



Figure 4 -Yearly electricity and heat consumption graphs



If users uploaded hourly data, they'll also be able to access more detailed graphs, that include hourly heat and electricity consumption and performance. An example of hourly data visualization is presented in the figure below: the black line in the graph represents the actual consumption, while the coloured bands help users to understand how close the actual consumption is to the hourly predicted consumption. If the black line coincides with the centre of the yellow band, it means that the actual consumption is equal to the predicted one, if the black line is in the green (red) area, the consumption is lower (higher) than then predicted, with the colours indicating quantitative deviations.

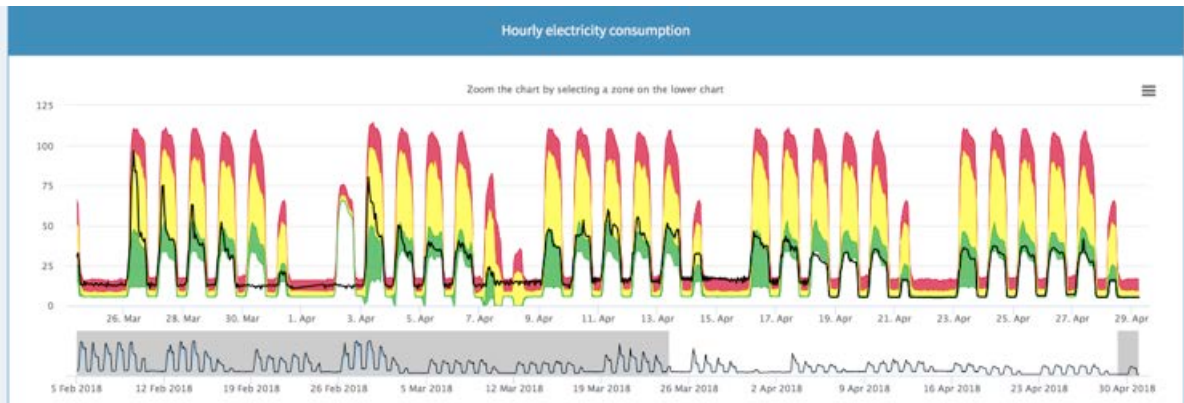


Figure 5 Hourly electricity consumption graph

**Communicate energy performance in a user-friendly manner**

In the user general dashboard, it is possible to visualize the energy performance of the different buildings managed by the user in a very simple and easy way, through icons and letters representing energy efficiency.

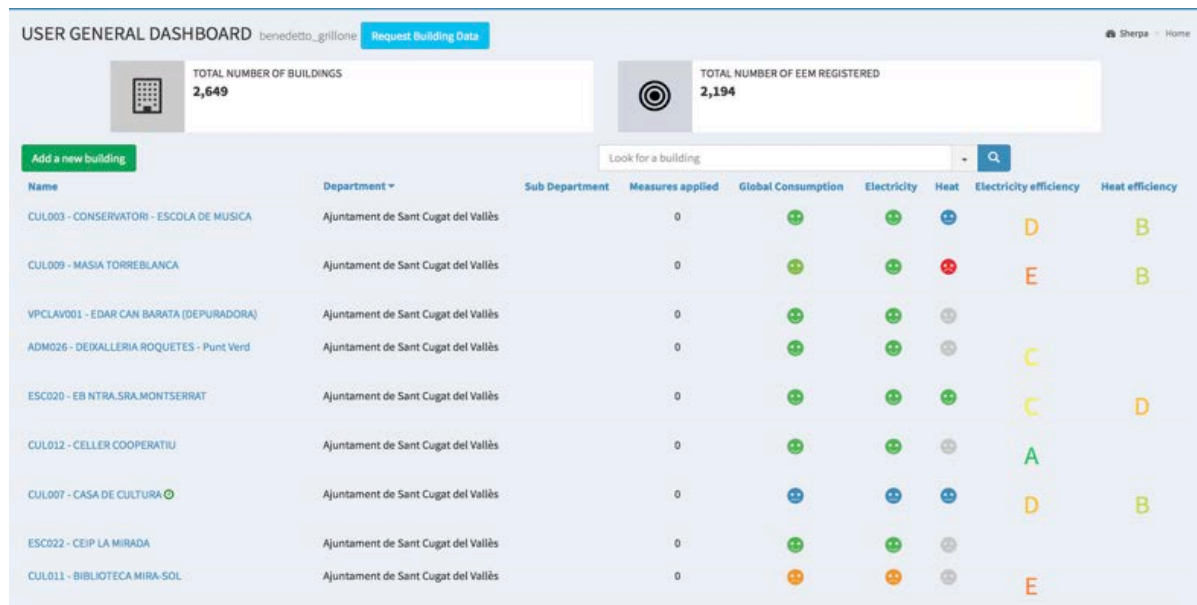


Figure 6 SHERPA general dashboard



Measures applied	Global Consumption	Electricity	Heat	Electricity efficiency	Heat efficiency
19				A	A

Figure 7 Detail of the general dashboard

Where the Smiley icons represent the comparison between the predicted consumption and the real consumption:

Consumption lower than the predicted by 8% or more	
Consumption lower than the predicted by 3 to 8%	
Consumption similar to the predicted (between 0 and 3% deviation)	
Consumption higher than the predicted by 3 to 8%	
Consumption higher than the predicted by 8% or more	

### Benchmark building energy use over time and to similar buildings

In the building dashboard, electricity and heat comparison graphics are accessible, through which the user can visualize the absolute and surface values of the building's energy consumption and compare it with the average and the most performing buildings in the database.

The energy performance indicator of the building is also shown in the form of a letter, according to the following rule, that takes as a criterion the area normalized energy consumption:

- A The building is in the 10<sup>th</sup> percentile with lowest consumption
- B The building is between the 10<sup>th</sup>. and the 25<sup>th</sup>. percentile
- C The building is between the 25<sup>th</sup>. and the 50<sup>th</sup>. percentile
- D The building is between the 50<sup>th</sup>. and the 75<sup>th</sup>. percentile
- E The building is between the 75<sup>th</sup>. and the 90<sup>th</sup>. percentile
- F The building is above the 90<sup>th</sup>. percentile





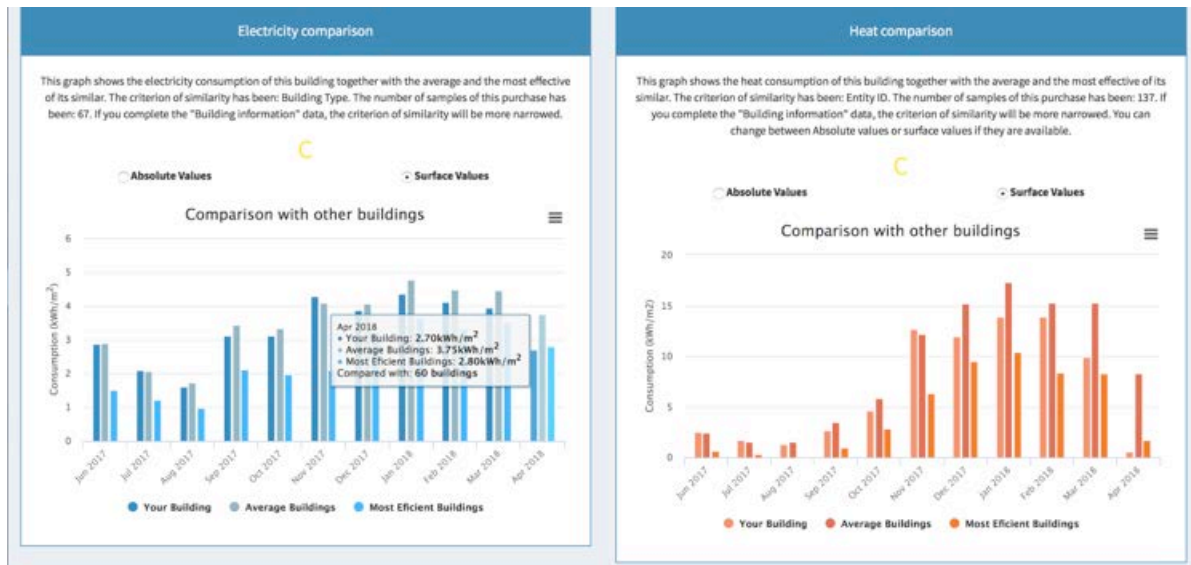


Figure 8 Electricity and heat benchmarking graphs

### Collect EEM applied together with the investments

Users can also access a tab with the list of all the EEM applied to the selected building, including information about the category and the description of each measure, as well as the total investment related with its application.

Date	Source	Description	Investment
01/02/14	Management	Monitoring of the equipment consumption	6,000.0
01/01/14	Heating	Shutdown of the heating system during weekends and holidays	None
01/01/14	Cooling	Others	None
20/12/13	Electricity	Reduce the temperature of the data processing center (DPC)	35,000.0
19/09/13	Heating	Implementation of an integral Building Management System	60,000.0

Figure 9 Energy efficiency measures summary for a specific building

### Facilitate peer learning and manage energy efficiency intervention plans

In the Recommendations section of the platform, users can access which were the most applied energy efficiency measures in other buildings, by category. Information about the expected investment and savings are also shown, helping building owners and energy managers to plan an effective intervention plan.



## Heating

- Replacement of heat production equipment a with more efficient one

Buildings in which it has been applied : 57

average investment : 35,643.02 EUR

expected investment : 528,955.42 EUR

- Change of single components in the heat production system ( burner, compressor, ...)

Buildings in which it has been applied : 8

average investment : 36,043.88 EUR

expected investment : 167,117.46 EUR

- Replacement of circulation pumps

Buildings in which it has been applied : 6

average investment : 5,165.50 EUR

expected investment : 15,959.41 EUR

## Cooling

- Cooling equipment partial change (compressor, exchanger , ...)

Buildings in which it has been applied : 9

average investment : 8,591.56 EUR

expected investment : 21,967.49 EUR

- Cooling system's working time rescheduling

Buildings in which it has been applied : 4

average investment : 1,158.50 EUR

expected investment : 2,009.58 EUR

- Distribution system balancing

Buildings in which it has been applied : 3

average investment : 20,266.67 EUR

expected investment : 61,967.92 EUR

Figure 10 Recommendations section of the SHERPA platform







#### 4. References

(Acouemo 2015) H. N. Akouemo Kengmo Kenfack, Data cleaning in the energy domain, Ph.D. thesis, Faculty of the 545 Graduate School, Marquette University, 00000 (2015). 546

URL <http://povinelli.eece.mu.edu/publications/papers/akouemo.pdf>

(Stuart 2011) Graeme Stuart. Monitoring Energy Performance in Local Authority Buildings, Doctoral thesis of De Monfort University, Leicester, UK, 2011.

(IPMVP) International Performance Measurement & Verification Protocol, Concepts and Options for Determining Energy and Water Savings, Volume I. Revised January 2012. [www.evo-world.org](http://www.evo-world.org)



## 5. Annexes

### 5.1. Description of the SHERPA API

#### 5.1.1. SHERPA Application API

The SHERPA Application API establishes the communication between final users or third-party systems and SHERPA system.

The SHERPA system has client-server architecture. The client communicates with the server using a RESTful API. The API can be easily adapted to further user necessities.

Inputs can be pushed in two ways:

- *Manually by final users: the user will be supported to upload files (e.g. CSV with consumption of last day) by SHERPA dashboard, in Data import section, and*
- *If pilot case or other organization has third party software with active APIs, will be able to push directly their data. They only need to enter their link in Data import section of the SHERPA dashboard.*

Outputs are the same than output resources that Big Data API provides (see below for more information).

#### 5.1.2. Big Data API

Whole communication between Big Data Analytics and SHERPA Application interfaces is done through the communication interface Big Data API. The description of usage and components can be found in the website: <https://api.sherpa.cimne.com/v1/docs/>

As the SHERPA Application API, the communication interface Big Data API is RESTful. The BigData API relies on stateless, client-server, and cacheable communication using the HTTPS protocol. And it is implemented using JSON codification.

The communication link is internally integrated and joint internally with the SHERPA system. The users do not directly connect to it and will only be able to see the communication interface SHERPA APP API.

A summary of the API implementation is based three points:

- *Establishment of communication methods,*
- *Definition of input resources,*
- *Definition of output resources*



### 5.1.3. REST methods

There are 4 methods that are implemented to use with input/output resources, although not all are always implemented because it depends on the type of resource (the allowed methods are specified in the documentation of each resource).

- **GET:** This command pushes data to the Big Data Analytics interface. Insert a new item or a list of items in a resource collection.
- **POST:** This command retrieves information from the Big Data Analytics interface. Retrieves an item or list of items of resource
- **PATCH:** This command updates the data in the Big Data Analytics interface.
- **DELETE:** This command deletes the information in the Big Data Analytics interface.

### 5.1.4. Input Resources

There are four different input resources: *Buildings*, *Contracts*, *Amon measures* and *Modelling units*. The description of each resource is provided below.

***Buildings:*** *Buildings* is the input resource used to obtain information relative to a specific building. It allows the four communication methods. The main fields are the following:

- **buildingId:** Required. Unique Id of the building
- **entityId:** Required. Entity to which the building belongs
- **address:** Information relative to the building location
- **data:** Information about building features like construction year, type of building, etc.
- **profile:** Information relative to description of the building user profile
- **system:** Information about building facilities
- **eem:** Information relative to Energy Efficiency Measures

***Contracts:*** *Contracts* get information about the tariff, period, and contracted power for each utility and the corresponding list of physical or virtual utility meters (*devices*) associated. As *devices* are defined as a list, it allows the flexibility to introduce several meters (electric, thermal, etc.), or maintains a historical list of *devices*. It allows the four communication methods. The main fields are the following:

- **contractId:** Required. Id used for each utility internally
- **buildingId:** Required. The ID of the building associated to the contract
- **tariffId:** Optional. The ID of the tariff applied to the contract
- **power:** Optional. The contracted power in the contract in [W]



- **dateStart:** Required. The date start of the contract
- **dateEnd:** The date end of the contract when needed (null is admitted)
- **devices:** List of devices associated to this contract with the corresponding *deviceId* (each meter) and *multiplier* and *dateStart* (and *dateEnd* when needed)

**Amon measures:** *Amon measures* is a "virtual" input resource that saves the consumption data from any type of energy meter. It allows the communication methods GET, POST and DELETE, but with some exceptions considered in the documentation. The main fields are the following:

- **deviceId:** Required. Id used for each meter
- **measurements:** It considers each *timestamp*, *type* and *value* of energy consumption taken by a meter
- **readings (used internally):** It defines the parameters (or readings) that can be measured by a meter

**Modelling units:** *Modelling units* is an input resource used to define the requirements on how to calculate the *Baseline* model. It allows the four communication methods. The main fields are the following:

- **modellingUnitId:** Required. Id used for each modelling unit (e.g. a building)
- **label:** Description of the modelling unit
- **location:** Postal code and country to find the nearest weather station
- **baseline:** Timestamp interval to generate the *Baseline* model
- **devices:** List of devices associated to the modelling unit with the corresponding *deviceId* (each meter) and *multiplier*

### 5.1.5. Output Resources

There are three different output resources: *Raw data*, *Data quality* and *Baseline*. They only allow the communication method GET. The description of each resource is provided below.

**Raw data:** *Raw data* represents the whole amount of consumption data without any data treatment. *Raw data* model shows the ordered time series of each meter. The main fields are the following:

- **deviceId:** Required. Id used for each meter
- **timestamps:** List of considered timestamps
- **values:** List of energy consumption values



**Baseline:** *Baseline* generates the consumption profile of the building according each type of energy (Electricity, Gas, Water, etc.). Initially, before calculating the *Baseline* model, a “data cleaning” module (to eliminate the data gaps) is executed for resampling the *Raw data*. Then, the *Modelling units*’ resource is applied through the sum of *devices* (with each corresponding to a *deviceId* and a *multiplier*) and weather information (based on exterior temperature gathered from the nearest weather station) in order to obtain the *Baseline* model with predicted values. This model will allow multiple visualizations for the user like consumption benchmarking against similar buildings, evaluation of building energy savings, or to evaluate the energy saving achieved by each of the energy efficiency measures – EEMs (or group of EEMs). The main fields are the following:

- **modellingUnitId:** Modelling unit Id
- **timestamps:** List of considered timestamps
- **values:** List of energy consumption values
- **p05:** Percentile 5 in the distribution of energy predicted values
- **p25:** Percentile 25 in the distribution of energy predicted values
- **p75:** Percentile 75 in the distribution of energy predicted values
- **p95:** Percentile 95 in the distribution of energy predicted values

