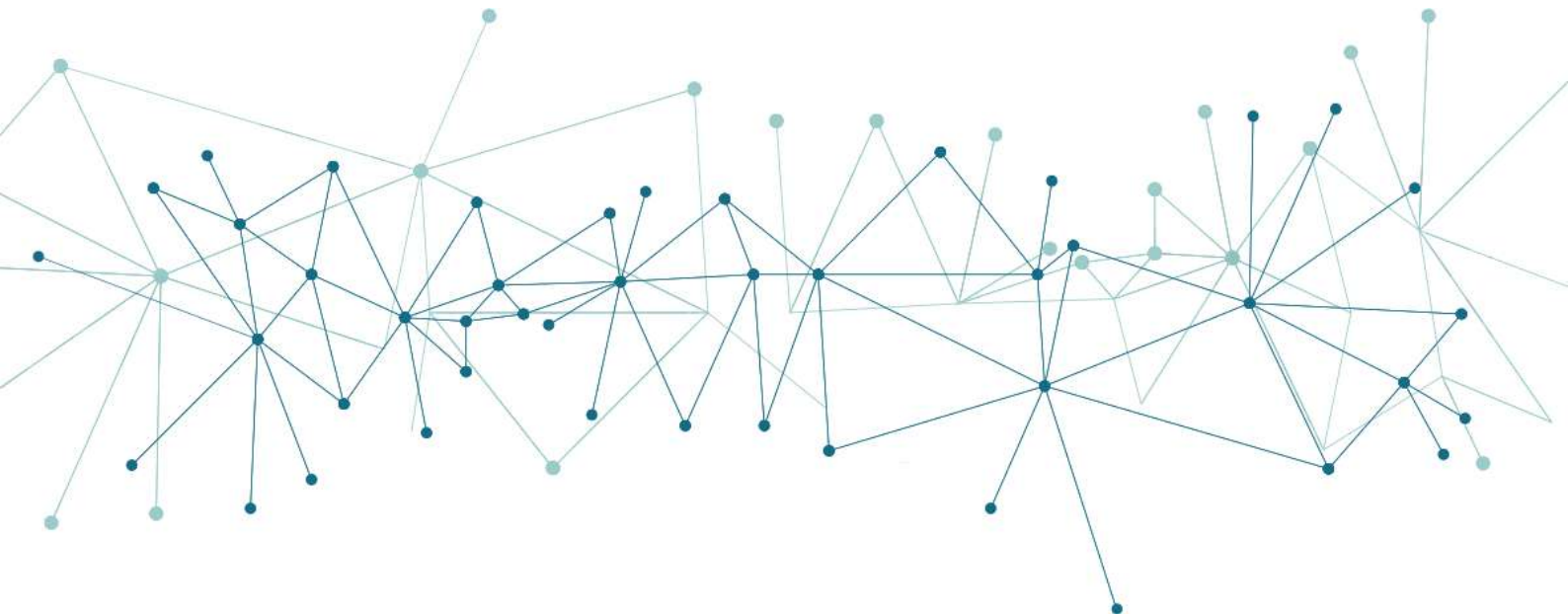




enabling new Demand REsponse Advanced, Market oriented and
secure technologies, solutions and business models

DELIVERABLE: D4.8 Interactive Visualization framework for improving DR strategies V2

**Authors: Paraskevas Koukaras (CERTH), Napoleon Bezas (CERTH),
Pashalis Gkaidatzis (CERTH), Thanos Mekras (CERTH), Dimosthenis
Ioannidis (CERTH), Christos Tjortjis (CERTH)**



Imprint

Interactive Visualization framework for improving DR strategies V2, April 2020

Contractual Date of Delivery to the EC: 30.04.2020

Actual Date of Delivery to the EC: 30.04.2020

Author(s):	Dimos Ioannidis (CERTH), Napoleon Bezas (CERTH), Paraskevas Koukaras (CERTH), Giannis Gotsos (CERTH), Paschalis Gkaidatzis (CERTH), Thanos Mekras (CERTH), Christos Tjortjis (CERTH)
Participant(s):	Contributors: Benjamin Hunter (TU), Fathi Abugchem (TU), Vladimir Vukovic (TU), Victoria Murcia (KIWI), Ugo Stecchi (ATOS), Javier Gomez (ATOS), Maria Lourdes Gallego Miguel (ATOS), Tommaso Bragatto (ASM), Francesca Santori (ASM)
Project:	enabling new Demand Response Advanced, Market oriented and secure technologies, solutions and business models (eDREAM)
Work package:	WP4 – Next generation DR Services for Aggregators and Customers
Task:	4.3 – Multi-level Visualization for enhanced user interaction
Confidentiality:	public
Version:	1.0

Legal Disclaimer

The project enabling new Demand Response Advanced, Market oriented and secure technologies, solutions and business models (eDREAM) has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 774478. The sole responsibility for the content of this publication lies with the authors. It does not necessarily reflect the opinion of the Innovation and Networks Executive Agency (INEA) or the European Commission (EC). INEA or the EC are not responsible for any use that may be made of the information contained therein.

Copyright

© <Centre for Research and Technology Hellas – CERTH, Thermi Thessaloniki – Central Directorate, 6th km Charilaou Thermi Rd>. Copies of this publication – also of extracts thereof – may only be made with reference to the publisher.

Table of Contents

List of Figures	4
List of Tables	5
List of Acronyms and Abbreviations	6
1 Introduction	8
1.1 Purpose.....	8
1.2 Relation to other activities	8
1.3 Structure of the deliverable.....	9
2 Graph Analytics Platform – General view	10
2.1 Context & Design	10
2.2 Front-End	12
2.3 Dataset Description	14
3 Graph Analytics Platform – detailed view	16
3.1 Prosumers geolocation view.....	16
3.2 KPIs view	18
3.3 Aerial survey view.....	20
3.3.1 Multi-purpose Dashboard	20
3.3.2 Augmented Reality aspect.....	23
3.3.2.1 EPSON Moverio BT-300 Smart Glasses	24
3.3.2.2 ThirdEyeX2 Mixed Reality Glasses	25
3.3.2.3 Vuzix Blade Smart Glasses	26
3.3.2.3 Toshiba dynaEdge AR100 Viewer	27
3.4 Trend analysis view.....	28
3.5 Multi-objective visual clustering for common behaviors	31
3.6 Multi-objective visual clustering with specific feature selection.....	34
3.7 k-partite visual clustering of common features using binning	37
3.8 Geographical analysis utilizing heatmaps.....	40
3.9 Geographical analysis utilizing virtual clusters	43
3.10 Load profiling.....	47
3.11 DSS optimal scheduling	49
4 Integration context	50
5 Conclusions	53
References	54

List of Figures

Figure 1: eDREAM pert diagram showing WP4 relations with other deliverables.	8
Figure 2: Navigation menu for multi-purpose dashboard.	11
Figure 3: eDREAM login view.	12
Figure 4: eDREAM sign up view.	13
Figure 5: Home view of user Aggregator.	13
Figure 6: Control view of user Prosumer.	14
Figure 7: Logical view of integrated visualization framework.	16
Figure 8: Screenshot of geolocation view.	17
Figure 9: KPIs statics with multiple Prosumers Ids.	19
Figure 10: KPIs statistics timestamp selection.	19
Figure 11: Aerial survey overview.	21
Figure 12: Aerial survey info of building popup.	21
Figure 13: Aerial survey energy class of a building.	22
Figure 14: Aerial survey building features, assets, environmental conditions.	22
Figure 15: Mobile SDK interface.	23
Figure 16: Windows platform version SDK interface.	23
Figure 17: onboard SDK interface.	24
Figure 18: Moverio BT-300 [13].	24
Figure 19: Mixed Reality Glasses [14].	25
Figure 20: Blade Smart Glasses [16].	26
Figure 21: Toshiba dynaEdge AR100 Viewer [18].	27
Figure 22: Trend Analysis output for 1 day.	29
Figure 23: Trend Analysis output for 3 days.	30
Figure 24: Trend Analysis output for 7 days.	30
Figure 25: Multi-objective Visual clustering for common behaviors for identifying common groups based on the data from the TERNI pilot site (examined features are considered the consumption (Ea) and the production (E0) in kWh).	33
Figure 26: Multi-objective Visual clustering for common behaviors feature choices.	33
Figure 27: Graph clustering using Multi-objective Optimization for Multimodal Visualization.	35
Figure 28: Graph clustering using Multi-objective Optimization for Multimodal Visualization, feature selection.	36
Figure 29: Graph clustering using Multi-objective Optimization for Multimodal Visualization, node feature and KPIs values.	36
Figure 30: k-partite visualization with binning of prosumer consumptions.	38
Figure 31: k-partite visualization with binning of prosumer consumptions close look-up.	39
Figure 32: Heatmap of prosumers energy consumption.	41
Figure 33: Heatmap of prosumers energy consumption choosing timestamp.	42
Figure 34: Heatmap of prosumers energy consumption choosing timestamp zoom in and prosumer info.	42
Figure 35: Visualization using clustering for extracting the flexibility for Virtual neighborhoods.	44
Figure 36: Visualization using clustering for extracting the flexibility for Virtual neighborhoods with soft zoom in.	45
Figure 37: Visualization using clustering for extracting the flexibility for Virtual neighborhoods with more zoom in.	45
Figure 38: Visualization using clustering for extracting the flexibility for Virtual neighborhoods and info about prosumer.	46
Figure 39: Clustering of prosumers historical load consumption class#0.	48
Figure 40: Clustering of prosumers historical load consumption class#1.	48
Figure 41: DSS output for day ahead optimal scheduling.	50
Figure 42: DSS output for optimal scheduling for a particular intraday time period.	50

List of Tables

Table 1: eDREAM front-end key users.....	12
Table 2: eDREAM Dataset Description.....	14
Table 3: Prosumers geolocation view characteristics.....	16
Table 4: KPIs view characteristics.....	18
Table 5: Aerial survey view characteristics.....	20
Table 6: EPSON Moverio BT-300 Specifications [13].....	24
Table 7: ThirdEye X2 Mixed Reality Glasses Specifications [15].....	25
Table 8: Vuzix Blade Smart Glasses Specifications [17].....	26
Table 9: Toshiba dynaEdge AR100 Viewer Specifications [18].....	27
Table 10: Trend Analysis view characteristics.....	28
Table 11: Multi-objective Visual clustering view characteristics.....	31
Table 12: Multi-objective Visual clustering with specific feature view characteristics.....	34
Table 13: k-partite visual clustering using binning view characteristics.....	37
Table 14: Geographical historic data Analysis using a heatmap view characteristics.....	40
Table 15: Geographical day-ahead data Analysis with virtual cluster generation view characteristics.....	43
Table 16: Load Profiling view characteristics.....	47
Table 17: DSS optimal scheduling view characteristics.....	49

List of Acronyms and Abbreviations

API	Application Programming Interface
AR	Augmented Reality
eDREAM	enabling new Demand Response Advanced, Market oriented and secure technologies, solutions and business models
DEG	Distributed Energy Generator
DER	Distributed Energy Resource
DR	Demand Response
DSO	Distributed System Operator
HMI	Human Machine Interface
RES	Renewable Energy Sources
VPP	Virtual Power Plant
WP	Work Package

Executive Summary

This document aims to report on the improvements made on the deliverable D4.4 titled “Interactive visualization framework for improving DR strategies V1” [3]. It is an expansion that is based on the activities of the Task 4.3 titled multilevel visualization for enhanced user interaction. The most important part of the visualization framework is the graph analytics platform which enables the aggregators as well as the prosumers to analyze large volume of data that is related with energy consumption. The aggregators need to be informed through monitoring and insights regarding the consumption/production profile data of the prosumers while the prosumers want to monitor their production/consumption. This deliverable is an enhanced version of the documentation manual provide it with the deliverable D4.4 presenting the improvements made during the development of their prototype version of eDREAM visualization platform.

The graph analytics platform aims to provide multiple factor/criteria analysis and detection of spatiotemporal patterns towards the improvement of aggregators portfolio. The logical view and the deployment view are presented in order to present and validate they are adopted techniques and models to be used addressing the challenges of the demand response sector. The high-level functionalities that this document describes are the following:

Visual clustering techniques such as k-partite, that offers grouping of prosumers for producing insights at the aggregator’s portfolio, which implements a multi-dimensional analysis and segmentation of the prosumers and their profiles. The distances that are formed in the graphs layer our dynamic in order to formulate what is clusters fulfilling different purposes. For example, provide visual data about flexibility, participation in the energy market or correlation of each prosumer KPIs within the overall context.

Furthermore, multi-objective analysis is used in order to perform multiple criteria decision making utilized mathematical optimization based on specific feature selection aiming to optimize multiple objective functions at the same time. As described in D4.4, multiple-objective optimization to make decisions considering the trade-off for example of a specific prosumer’s production and consumption at a given timestamp. More features like consumption or production can be present.

To offer the best possible visualization experience within the context of the aforementioned purposes of this deliverable, we employ various types of charts (bar charts, radar charts, graphs etc.) in multiple views. The output should be a multi-purpose HMI that produces useful and conceivable information for the end-user.

1 Introduction

1.1 Purpose

This report provides an overview of the work done in the direction of updating the work carried out during the Deliverable 4.4 [1], which referred to the definition of models and techniques for implementing an interactive visualization framework for improving demand response strategies for the end users (DSO/Aggregator/User). The interactive visualization HMI should be able to ease the monitoring and information retrieval for the end user regarding the flexibility of distributed energy prosumers considering various time frames and granularities. This deliverable essentially represents the progress regarding the HMI development after adapting, integrating, tuning and assembling various views with different purposes in order to generate a single and unified application.

The work has been done in relation with eDREAM project Task 4.3 “Multi-level Visualization for enhanced user interaction” which is logically linked with Task 3.1 “Prediction Methods for Electricity Consumption/Production Forecasting” part of Work Package (WP3) “Techniques for DR and Energy Flexibility Assessment”.

This document will report on the integration of the framework with the external generic optimization tools. The framework will be able to recommend the customers for additional services they can experience for better monitoring and further smartening and revenue generation of their energy assets and participation in DR programs. The success criteria should include the following aspects: Framework integration results, interoperability mechanisms and recommendation usage process.

1.2 Relation to other activities

WP4 uses the output from WP2 in terms of requirements and use case definition while aiming to design the eDREAM envisioned next generation of Demand Response services both in a classical, centralized approach as stated by the WP4 purposes which considers the main models and techniques of WP3 as shown in Figure 1.

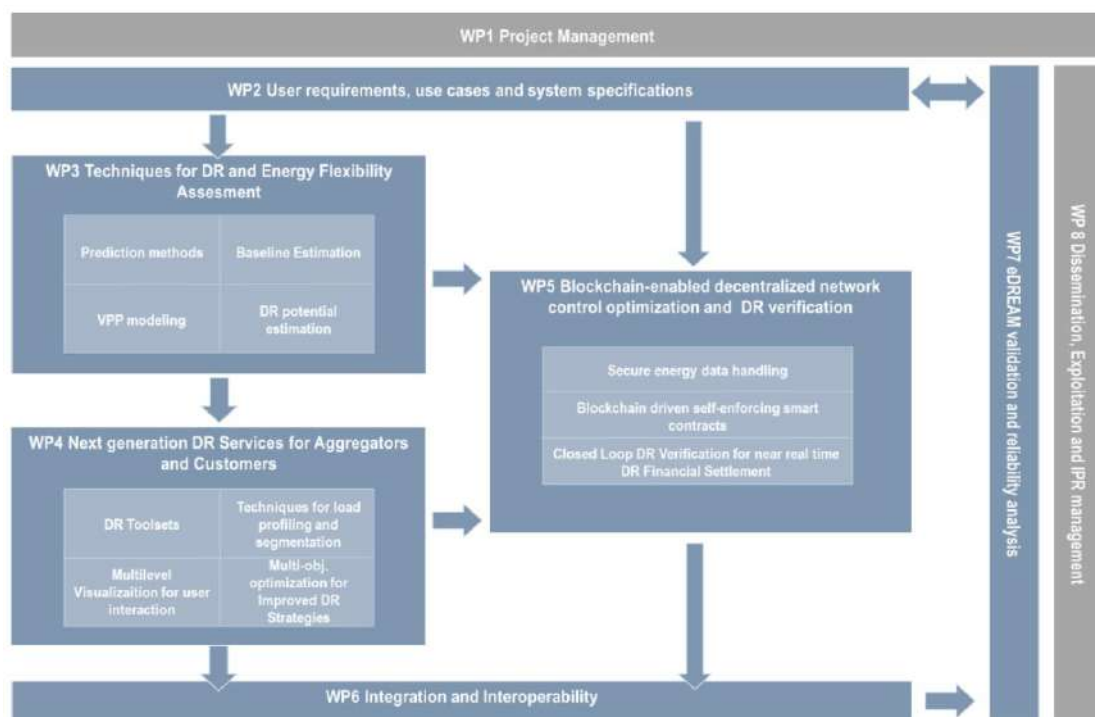


Figure 1: eDREAM PERT diagram showing WP4 relations with other deliverables.

1.3 Structure of the deliverable

- Section 1 presents introductory concepts, the purpose, as well as the relation to other activities of this deliverable;
- Section 2 describes the eDREAM Graph Analytics Platform and is split into three subsections. The first subsection refers to the general context and design of the platform, the second showcases some basic views of the front-end part and the third provides an overview of the datasets that are being used;
- Section 3 presents the logical view and the visual analytics techniques that are being used in order to fulfil the functional requirements of the Visual Analytics Platform. These are presented in tabular way showcasing 11 views that act as independent components unified under the same multi-purpose dashboard.
- Section 4 offers a description of the Integration process regarding augmented reality tools.
- Section 5 is dedicated to the collection and presentation of the summary of the tools/software that were used for the development of the visual analytics platform.
- Section 6 is devoted to the conclusions of this deliverable.

2 Graph Analytics Platform – General view

This section provides a detailed presentation of all the aspects of the graph analytics component including the various views that are implemented. This includes the presentation of all interactive visual characteristics that are necessary for the HMI. It also presents all the required visualization aspects and exploration of datasets for the eDREAM project's pilot sites. These datasets contain different kinds of energy measurements that can produce insights for prosumer's consumption, production habits and flexibility values. The graph analytics platform is a tool that offers multiple user interactive and functionality coordinated views aiming to provide a complete tool for the Aggregators and the DSO with various supportive visual aspects regarding the data that are related with their customers (prosumers). The implementation of these views aims to provide efficient visual exploration of multiple data aspects at the same time.

The Graph Analytics platform was developed taking into consideration the high requirements for scalability that are being exploited according to the Use Cases. It is quite common that the energy companies are compelled to handle very large amount of data that is associated with their clients (prosumers). While companies have their portfolio of clients increased the management process becomes more difficult. Eventually they handle big data with big data analytics issues. One of such issues is the identification of patterns in very large volumes of data. At the same time the HMI and its process of interaction can become a slower. This is an issue that we tried to tackle during the implementation process of the graph analytics platform, namely the great need for scalability and fast HMI considering the increasing energy market requirements utilizing DEG, DER, DR, RES, VPP etc. capabilities.

2.1 Context & Design

This section outlines the main features of the interactive visualization framework as toolsets developed and delivered as outcomes of the activities of T4.3 and the natural expansion of the work achieved and described in Deliverable 4.4 [1]. Therefore, there is information that is being reported in this document that has been already provided in Deliverable 4.4 which is normal since the latter contains the context of the design and this one the development and deployment of the concepts.

The Graph Analytics Platform is being continuously updated and implemented based on the user and market requirements that are reported on the outputs of WP1 (Project Management) and WP2 (User requirements, use cases and system specification).

This chapter offers a textual description of the toolset for the interactive visualization framework. Each tool is reported as a standalone view within the visualization platform. The user is able to navigate to the various views through a side menu (Figure 2).

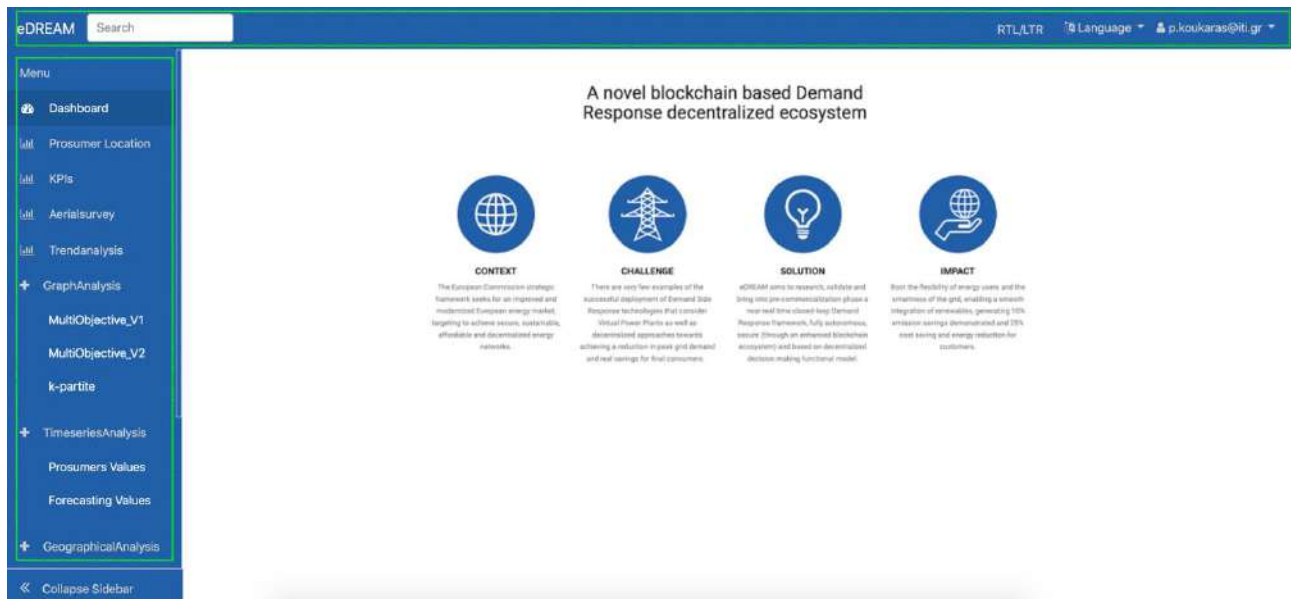


Figure 2: Navigation menu for multi-purpose dashboard.

According to the Context and Design the following functionalities are being included in the form of various deployed views within the multipurpose visualization platform and reported with example parameters:

- *Prosumers geolocation* that allows to interact with a map of the prosumers at one (e.g. Terni) or more pilot areas which uses dummy coordinates. (in the multipurpose visualization platform navigation menu is referred as “*Prosumer Location*”).
- *KPIs* which showcases the comparison of multiple user KPIs such as CO₂_reduction or Demand Flexibility (in the multipurpose visualization platform navigation menu is referred as “*KPIs*”).
- *Aerial survey* for checking characteristics of multiple building within the pilot areas (in the multipurpose visualization platform navigation menu is referred as “*Aerialsurvey*”).
- *Trend analysis output* for a specific timeslot (in the multipurpose visualization platform navigation menu is referred as “*Trendanalysis*”).
- *Multi-objective visual clustering for common behaviors* for a specific timeslot showing the individual and merged graphs based on specific prosumer feature selection (in the multipurpose visualization platform navigation menu is referred as “*Multi-Objective analysis V1*”).
- *Multi-objective visual clustering with specific feature selection* that allows feature selection to visualize prosumers with similar feature values (in the multipurpose visualization platform navigation menu is referred as “*Multi-Objective analysis V2*”).
- *k-partite visual clustering of common features using binning* that bins the users according to specific features based applying a datetime filter (in the multipurpose visualization platform navigation menu is referred as “*k-partite*”).
- *Geographical analysis utilizing heatmaps* that produces a heatmap of consumptions according to user datetime request (in the multipurpose visualization platform navigation menu is referred as “*GeographicalAnalysis (Historic)*”).
- *Geographical Analysis utilizing virtual clusters* that forms clusters of close vicinity aggregated consumptions of multiple prosumers based on historic data (in the multipurpose visualization platform navigation menu is referred as “*GeographicalAnalysis (Day-Ahead(hourly))*”).
- *Load Profiling* that allows to find out the number of clusters that the prosumers are grouped into, based on the feature of load consumption. (in the multipurpose visualization platform navigation menu is referred as “*OptimalScheduling (Clustering Analysis)*”).

- *DSS optimal scheduling* that allows to check out the participation rate of prosumer's generation and the load shedding for every time slot to achieve the minimization of the operational cost. (in the multipurpose visualization platform navigation menu is referred as "*OptimalScheduling (Barchart Analysis)*").

2.2 Front-End

In this sub-section, we present the functionalities supported by the current version of the Graph Analytics Platform. These functionalities will be enhanced and enriched until the end of the project. Some Figures follow that present different states during the development in order to pitch the various views of the Front-End of the Graph Analytics platform.

The key users of the eDREAM platform are grouped in two type groups as reported in the following table [2]:

Table 1: eDREAM front-end key users.

Energy Sector	End users
Energy retailers	Prosumers
DSOs	Facility managers & owners
Distributed Generation Providers	System operators
Energy Aggregators and brokers	Commercial and Residential Customers
ESCOs	Stakeholders at the Pilot Sites
Technology Providers	General Public
Scientific community	

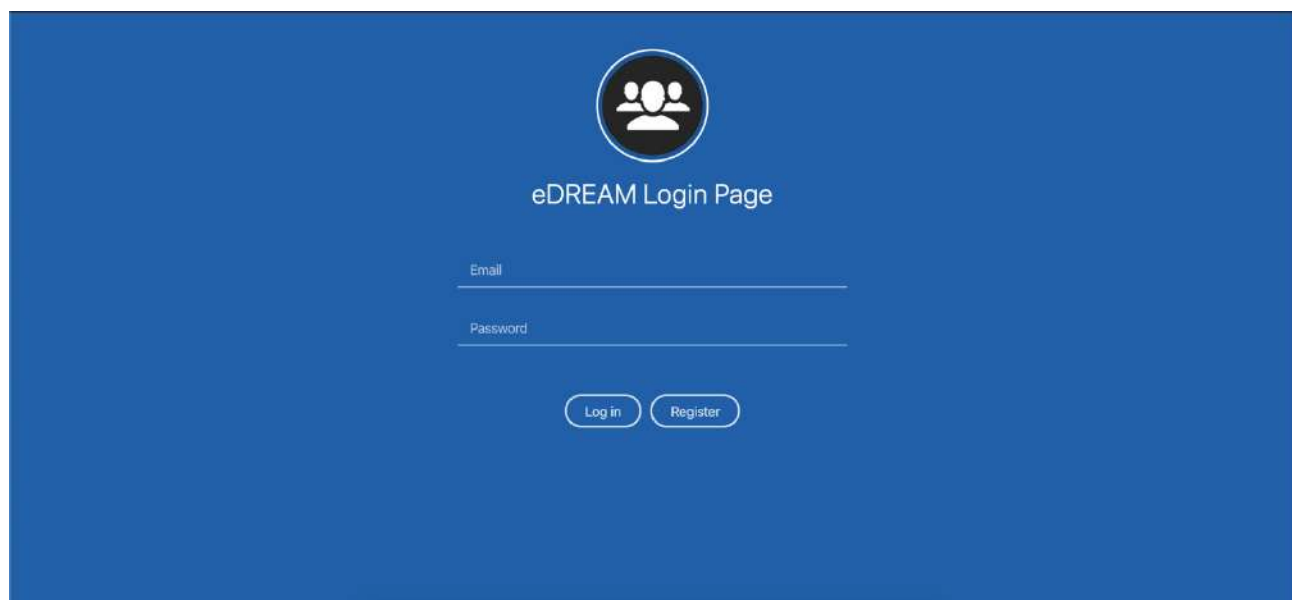



Figure 3: eDREAM login view.




eDREAM Sign up Page

First Name
 Last Name
 Email
 Password
 Repeat Password

Figure 4: eDREAM sign up view.

According to Table 1, the front-end platform will allow 2 type groups to access its contents. The energy sector experts (DSO, Aggregator) and the end users (Prosumers). Therefore, at the current development phase we distinguish two test users that each one of them represents each of the two groups.

The users have access to specific functionalities according to their type group, as depicted in the following 2 figures:



Figure 5: Home view of user Aggregator.

The figure above illustrates the upcoming DR events and the fluctuation of marginal system price during the DR events (completed, current, upcoming).

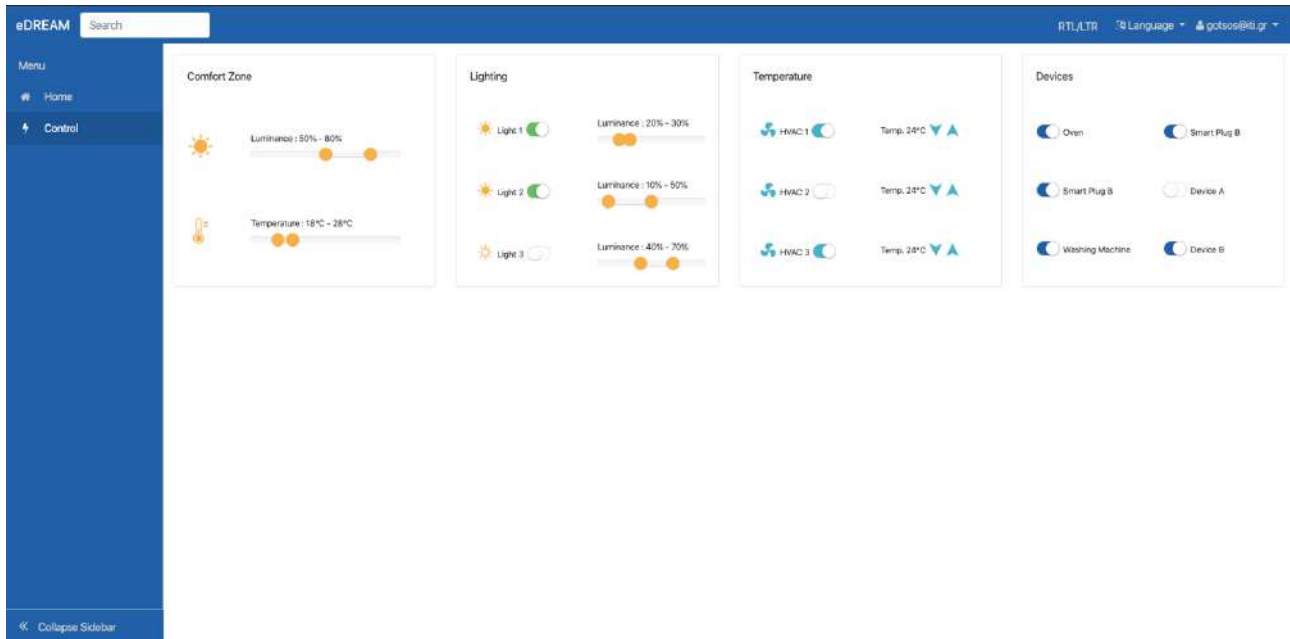


Figure 6: Control view of user Prosumer.

The figure above shows that the end user has access to his assets by monitoring his comfort zone levels, lightning, temperature and turn on or off his/her devices and having the ability adjust them.

2.3 Dataset Description

In this section, we present the datasets that were used during the development and operation of the graph analytics platform. Whenever timeseries are utilized, the timestamp is in coordinated universal time (UTC), in which way they are stored. However, whenever time data are to be utilized, the local time is presented. For each view of the UI we describe the dataset that we used on the following table:

Table 2: eDREAM Dataset Description.

Views	Data Description
Prosumers Geolocation	The dataset consists of: <ul style="list-style-type: none"> ➤ ID of each prosumer ➤ Coordinates of the prosumers (a pair of latitude, longitude)
KPIs	Different features of prosumer: <ul style="list-style-type: none"> ➤ Historical consumption of prosumers (Ea) ➤ Historical generation of prosumers (E0)
Aerial Survey	<ul style="list-style-type: none"> ➤ Coordinates of prosumers (latitude, longitude) ➤ Weather data (temperature, humidity, wind speed, cloud coverage, etc.)
Trend Analysis	<ul style="list-style-type: none"> ➤ Historical generation data of prosumers

Multi-objective visual clustering for common behaviors	<ul style="list-style-type: none"> ➤ Historical load consumption data of prosumers (Ea) ➤ Historical generation data of prosumers (E0)
Multi-objective visual clustering with specific feature selection	<ul style="list-style-type: none"> ➤ Historical load consumption data of prosumers (Ea) ➤ Historical generation data of prosumers (E0)
K-partite visual clustering of common features using binning	<ul style="list-style-type: none"> ➤ Historical load consumption data of prosumers (Ea) ➤ Historical generation data of prosumers (E0)
Geographical Analysis	<ul style="list-style-type: none"> ➤ Coordinates of prosumers (latitude, longitude) ➤ Historical load consumption data of prosumers (Ea) ➤ Historical generation data of prosumers (E0)
Load profiling	<ul style="list-style-type: none"> ➤ Historical load consumption data of prosumers (Ea)
DSS Optimal Scheduling	<ul style="list-style-type: none"> ➤ Aggregated load forecast ➤ Maximum generation capacity of each prosumer ➤ Cost of generation ➤ Cost of load shedding

3 Graph Analytics Platform – detailed view

This section refers to the logical view of the Graph Analytics Platform. The scope of this view is to present the decomposition of the system into the functional components of which it consists of. The logical view of the interactive visualization framework is concerned with the overall functionality that the system provides to the end-users (Aggregators, DSO, Prosumers). Figure 7 shows the logical view of the integrated visualization framework according to WP4.

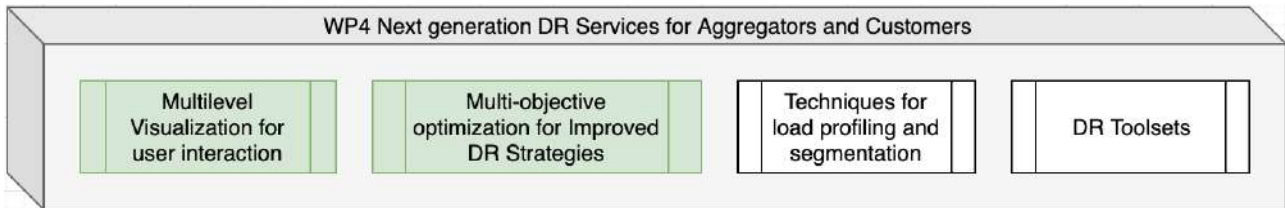


Figure 7: Logical view of integrated visualization framework.

The following sections offer a detailed presentation of the various views of the integrated visualization framework platform at the current state of the development. The visual analytics techniques are merged into one *multi-purpose dashboard* that aims to deliver state of the art technologies integrating energy monitoring capabilities by analyzing prosumer data providing information about past and real-time prosumer energy consumption.

3.1 Prosumers geolocation view

The prosumer geolocation view is one of the simplest views of the multi-purpose dashboard as it allows the user to interact with a map that has landmarks of the available prosumers. Some functionalities are supported such as *zoom in* and *zoom out* as well as basic interactions with the landmarks, for example on click the user can get some basic data about each of the prosumers.

Table 3: Prosumers geolocation view characteristics.

	Description
Functionality description	The navigators of this view can interact with a map of the prosumers at one (e.g. Terni) or more pilot areas which uses dummy coordinates.
Visualization capabilities	This view offers a simple representation of the prosumers' location. At the same time, we can see the exact coordinates of each prosumer by clicking on his/her location on the map. Also, there are zoom in and zoom out capabilities.
Technical specifications	<p>This view is developed by utilizing the Angular 8 framework, NodeJS and MySql-MariaDB database and the Mapbox GL JS JavaScript library.</p> <p><i>Server Hardware</i> CPU: Intel(R) Core(TM) i5-6600 CPU @ 3.30GHz Memory: 16GB</p> <p><i>Software</i> Operating System: Windows 10 Pro</p>

	MySQL version: 10.1.19-MariaDB NodeJS version: 12.0.0 Mapbox GL JS version: 1.2.0 Sequelize Framework Version: ^5.8.5
The external dependencies include	There are no external dependencies.
Input data	The input data is a json entry that contains the coordinates of each prosumer. For example, the input data of a single prosumer is: <pre>{ "type": "FeatureCollection", "features": [{ "type": "Feature", "geometry": { "type": "Point", "coordinates": [12.647669, 42.573196] }, "properties": { "ID": 1 } }] }</pre>
Output data	The output data is a location on the interactive map of this view.
Implementation features and delivery	Currently the tool is hosted by a web server and is released as an Angular 8 project that contains multiple components. The source code is versioned on the eDREAM gitlab repository.

Some screenshots of the tool are reported.

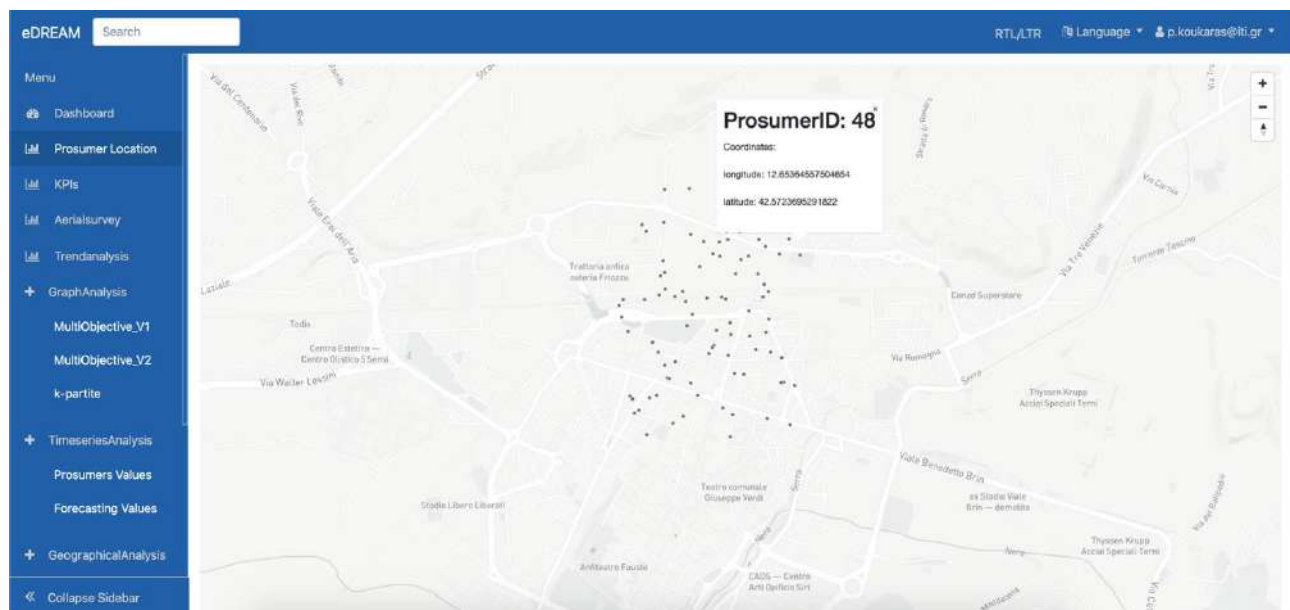


Figure 8: Screenshot of geolocation view.

3.2 KPIs view

This view of the multi-purpose dashboard allows the user to interact with a radar chart that visualizes values of KPIs for the available prosumers. Many functionalities are supported such as adding prosumers or removing prosumers for values' comparison as well as the ability to compare each of the selected prosumers with the overall min, mean, max levels of the visualized KPIs.

Table 4: KPIs view characteristics.

	Description
Functionality description	The navigators of this view can interact with a radar chart that shows various KPIs.
Visualization capabilities	The users can monitor various KPIs from a user specified number of different user as well as compare each value with the requested metric (min, mean, max) of all the aggregated values.
Technical specifications	<p>This view is developed by utilizing the Angular 8 framework, NodeJS and MySql-MariaDB database.</p> <p><i>Server Hardware</i> CPU: Intel(R) Core(TM) i5-6600 CPU @ 3.30GHz Memory: 16GB</p> <p><i>Software</i> Operating System: Windows 10 Pro MySql version: 10.1.19-MariaDB NodeJS version: 12.0.0 Sequelize Framework Version: ^5.8.5</p>
The external dependencies include	There are no external dependencies.
Input data	<p>The input data is a json entry that contains the KPIs from each prosumer</p> <p>For example, the input data of a single prosumer is:</p> <pre>[{"prosumers_id":1,"id":1,"datetime":"2017-04-27 09:00:00","start_coords_x":"42.57","start_coords_y":"12.65","Ea_kWh":"27.0","E0_kWh":"0.0","Demand_Flexibility_kWh":"70.0","CO2_reduction_gCO2/kWh":"70.0","createdAt":"2020-02-17T11:41:59.000Z","updatedAt":"2020-02-17T11:41:59.000Z"}].</pre>
Output data	The output data is forming the radar chart of this view. The aggregated data for metrics (min, mean, max) is calculated by on the front end, depending on the timestamp of the user choice.
Implementation features and delivery	<p>Currently the tool is hosted by a web server and is released as an Angular 8 project that contains multiple components.</p> <p>The source code is versioned on the eDREAM gitlab repository.</p>

Some screenshots of the tool are reported.

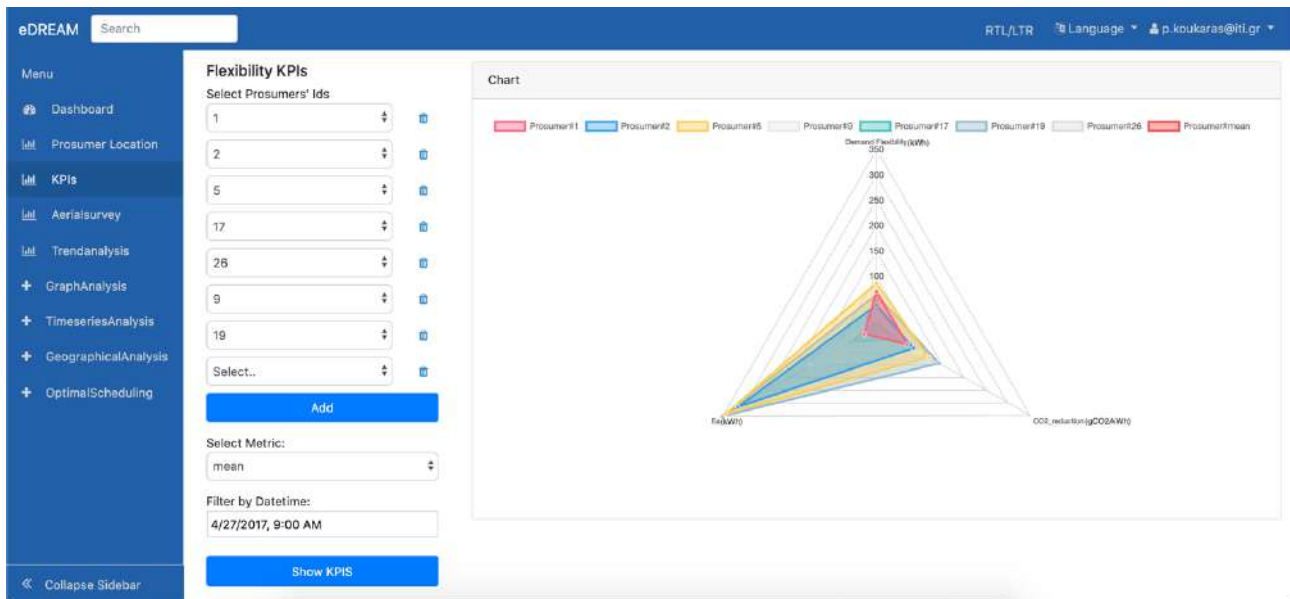


Figure 9: KPIs statics with multiple Prosumers Ids.

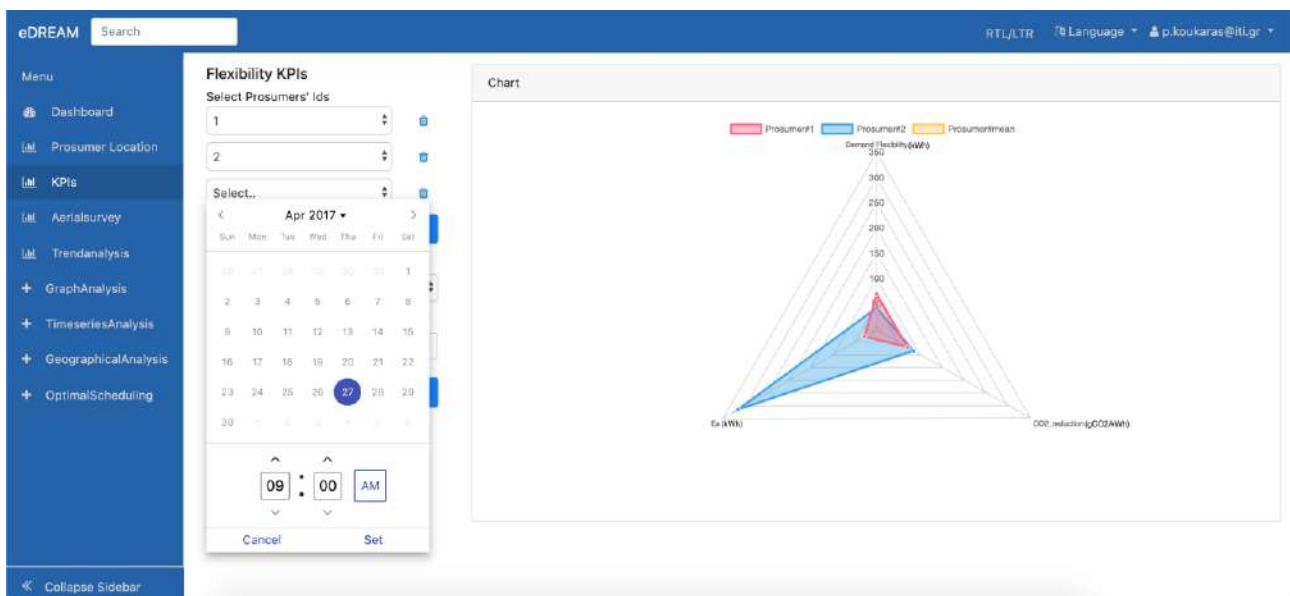


Figure 10: KPIs statistics timestamp selection.

3.3 Aerial survey view

3.3.1 Multi-purpose Dashboard

This view of the multi-purpose dashboard allows the user to interact with a map that contains building locations associated with the eDREAM pilot sites. Functionalities such as zoom in and zoom out on the map are supported as well as on click events that show information about the areas or the buildings selected. The information that the user can get is relevant with energy building classes, weather data, number of PVs etc.

Table 5: Aerial survey view characteristics.

	Description
Functionality description	The navigators of this view can interact with a map depicting the pilot sites' buildings and get an estimate of their energy efficiency.
Visualization capabilities	This view shows different energy features of the buildings. These features are Orientation, Thermal Index, Rood Surface, Windows, Energy Class etc.
Technical specifications	<p>This view is developed by utilizing the Angular 8 framework, NodeJS and MySql-MariaDB database and the Mapbox GL JS JavaScript library.</p> <p><i>Server Hardware</i> CPU: Intel(R) Core(TM) i5-6600 CPU @ 3.30GHz Memory: 16GB</p> <p><i>Software</i> Operating System: Windows 10 Pro MySql version: 10.1.19-MariaDB NodeJS version: 12.0.0 Sequelize Framework Version: ^5.8.5</p>
The external dependencies include	There are no external dependencies.
Input data	<p>The input data is a json entry that contains the characteristics of the buildings on the specific area as well as some input about weather conditions. For clarification reasons, the measuring units of the labels on the following input data are:</p> <p>temp, temp_min, temp_max, feels_like: Celsius scale pressure: atmospheric pressure, in Hectopascal humidity: relative air humidity, in %. visibility: distance of direct visibility, METAR, in meters. wind: speed in meters per second and direction in degrees. clouds: cloudiness, in %.</p> <p>Dt, sunrise, sunset: Time of measurement, Sunrise time, Sunset time in Unix Timestamp timezone: Shift in seconds from UTC name: Area name cod: internal parameter</p> <p>For example, the input data of the weather condition is:</p>

	<pre>{"coord":{"lon":22.99,"lat":40.57},"weather":{"id":800,"main":"Clear","description":"clear sky","icon":"01d"},"base":"stations","main":{"temp":16.14,"feels_like":11.4,"temp_min":14.44,"temp_max":17.78,"pressure":1021,"humidity":41,"visibility":10000,"wind":{"speed":4.6,"deg":200},"clouds":{"all":0},"dt":1587050551,"sys":{"type":1,"id":6658,"country":"GR","sunrise":1587008842,"sunset":1587056864},"timezone":10800,"id":6779072,"name":"Néa Kríni","cod":200}</pre>
Output data	The output data is forming the map along with the building. The view is updated accordingly, depending on the users' navigation choices.
Implementation features and delivery	Currently the tool is hosted by a web server and is released as an Angular 8 project that contains multiple components. The source code is versioned on the eDREAM gitlab repository.

Some screenshots of the tool are reported.

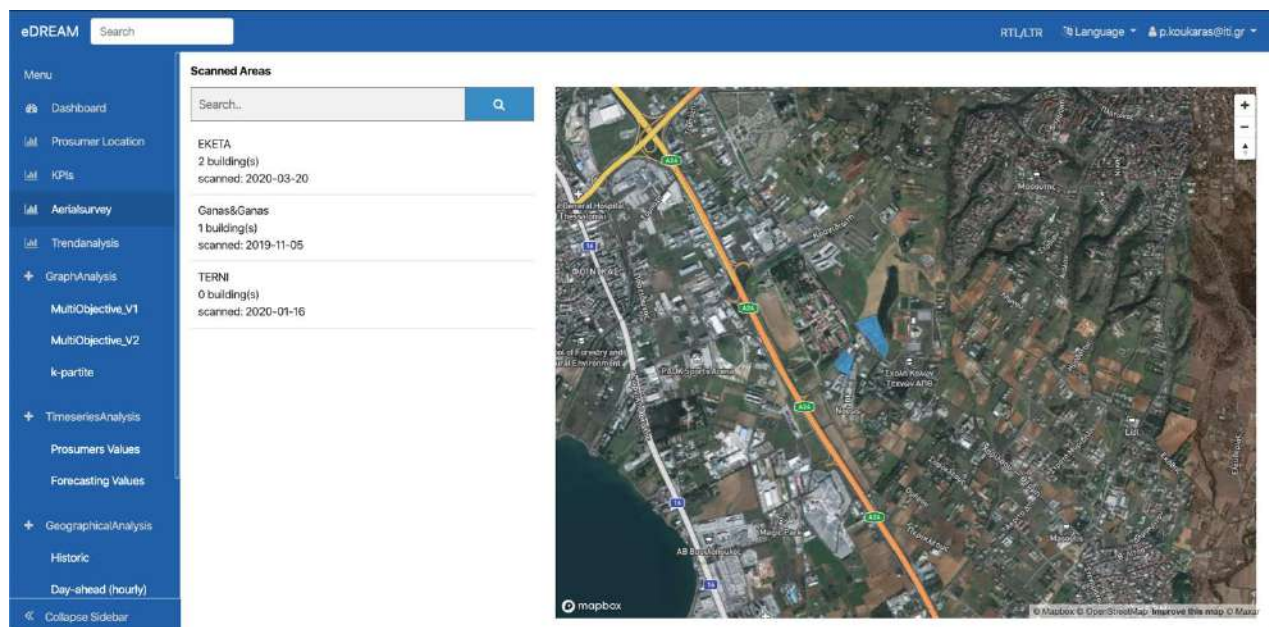


Figure 11: Aerial survey overview.



Figure 12: Aerial survey info of building popup.



Figure 13: Aerial survey energy class of a building.

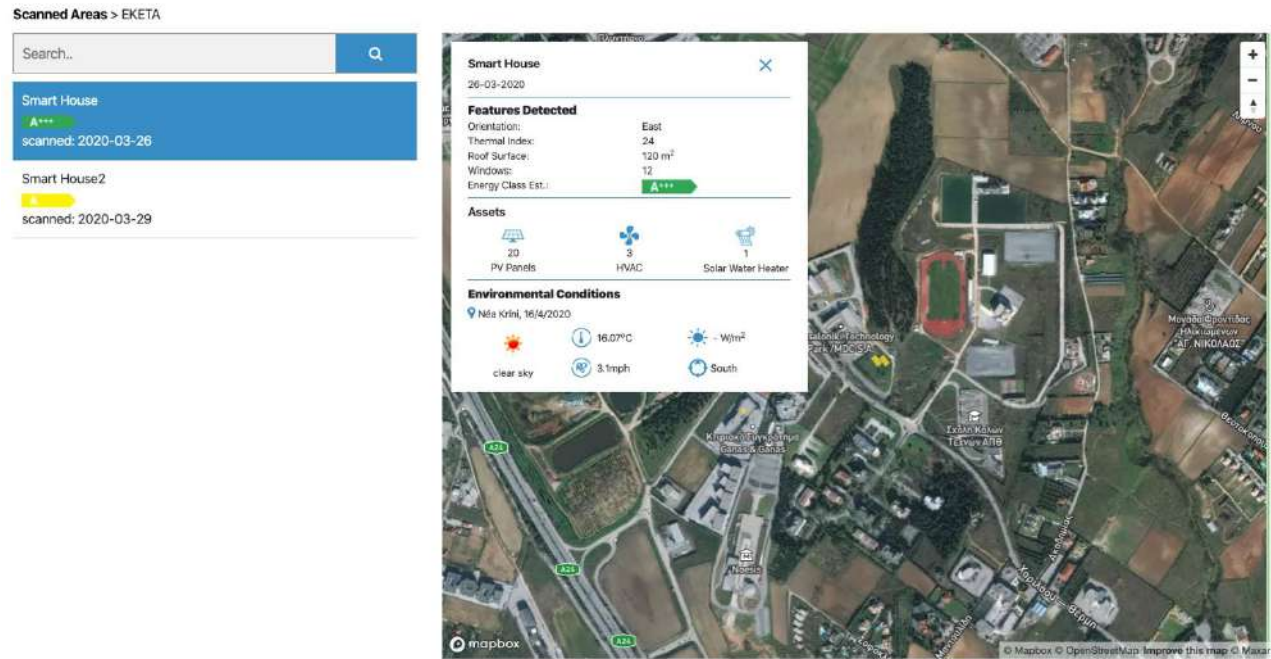


Figure 14: Aerial survey building features, assets, environmental conditions.

3.3.2 Augmented Reality aspect

Within the framework of the eDREAM project, in order to enhance the visualization experience of the end user during an aerial survey and Augmented Reality aspect has been implemented. Various Application Programming Interface (API), and/or Software development Kits (SDKs) have been identified and that either commercially available, or open source, such as DroneKit [5] that offers an API that allows developers to create Python apps that communicate with vehicles over MAVLink [6]. This API is open source and has proven a community driven project and also supports drone that communicate directly with MAVLink. There is also the Dronedeploy [7], that is open source and Flytbase [8] that is an open-source SDK compatible with all major drones commercially available (DJI, Ardupilot, PX4) and hardware platforms, including Intel, Qualcomm, Nvidia and Samsung. Ardupilot [9] is also another possible solution that provides hardware and software solutions, has an opensource autopilot software and supports the use of Raspberry Pi implementations. UgCS [10] is an easy to use software for planning and flying the unmanned aerial vehicle (UAV), that is to conduct the aerial survey. DJI itself provides several types of open source DSKs, though DJI drone apps can be developed only through DJI SDKs [11]. Since the drone that has been used so far is the DJI Matrice 200 [12], built by DJI, the DJI framework has been utilized as the most compatible.

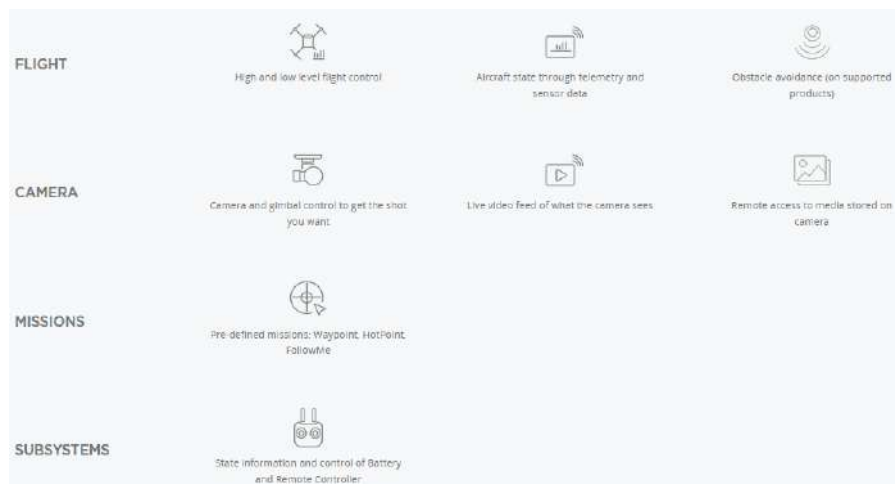


Figure 15: Mobile SDK interface.

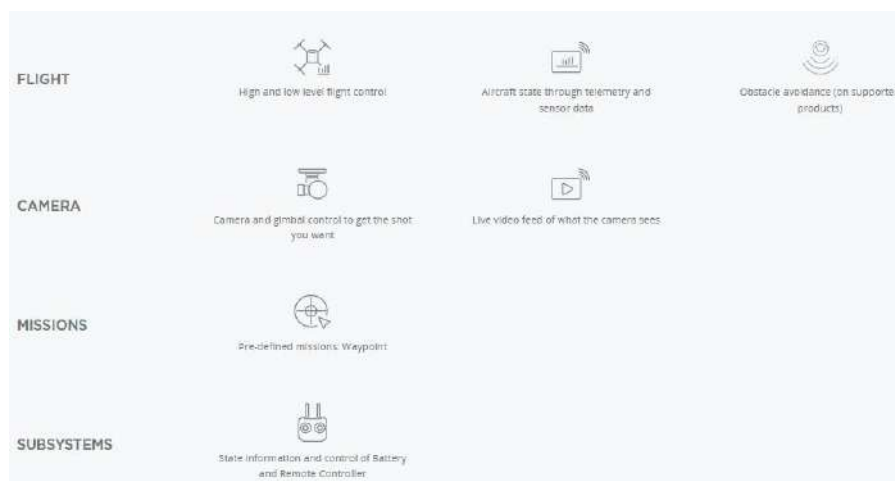


Figure 16: Windows platform version SDK interface.

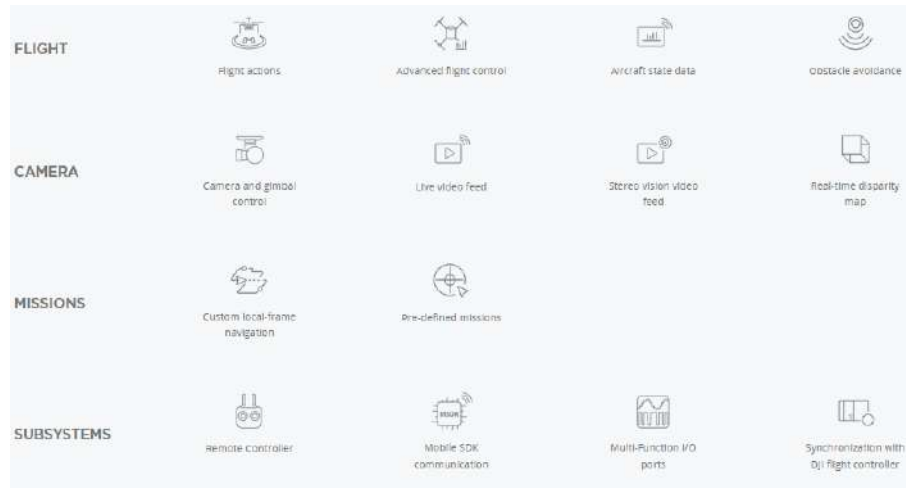


Figure 17: onboard SDK interface.

Regarding the specific features and functionalities of the AR tools, they present the same ones as those depicted in the multi-purpose dashboard, barring the map view. This means that the visualization is up to the building level, where the DR assets of the building are presented in high-real-time.

Regarding the hardware tools, apart from those already demonstrated in the previous version of this deliverable, that is Deliverable D4.4 [1], various other AR hardware tools have been identified, that are presented in the following sections.

3.3.2.1 EPSON Moverio BT-300 Smart Glasses

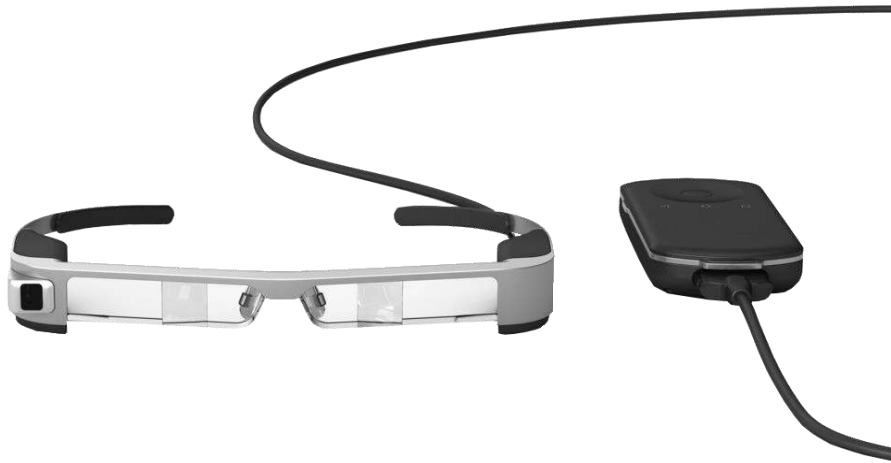


Figure 18: Moverio BT-300 [13].

The Moverio BT-300 features Epson's silicon-based OLED (organic light emitting diode) digital display technology, making the device one of the lightest binocular see-through smart glasses on the market with an OLED display quality. Specifications of the EPSON Moverio BT-300 Smart Glasses are presented in the table below.

Table 6: EPSON Moverio BT-300 Specifications [13].

Weight Headset	69 g (without light Shielding / without harness) (without cables)
Operating Temperature	5 ° C - 35 ° C, 41 ° F - 95 ° F, 20 % - 80 % Humidity
Battery Life	6 hrs (Video mode with Android at 25°)
Dimensions: Headset	191 mm x 178 mm x 25 mm (D x W x H)
Display Device Type	Si-OLED (Silicon - Organic Light-Emitting Diode)
Display size	0.43-inch-wide panel (16:9)

Pixel Number	921,600 pixels (1,280x 720) x RGB
Field of View	Approx. 23 °
Wireless LAN	IEEE 802.11a/b/g/n/ac with WiFi Direct and WiFi Miracast (Source/Sink)
Bluetooth	Bluetooth Smart Ready
microUSB	USB 2.0

3.3.2.2 ThirdEyeX2 Mixed Reality Glasses



Figure 19: Mixed Reality Glasses [14].

ThirdEye Gen X2 are augmented reality smart glasses which are compatible with AR and VR applications. Intended for professional and educational uses, these AR smart glasses can share the user's point of view with a remote person via video communication and live audio for remote assistance. They are made by ThirdEye Gen, a tech manufacturer based in the US. They boast an excellent battery life of 8 hours, but the user can also switch to external batteries. The below table presents the specifications of ThirdEye X2 Mixed Reality Glasses.

Table 7: ThirdEye X2 Mixed Reality Glasses Specifications [15].

Field of View	~ 42 Degrees
Resolution (pixels)	1280 x 720 HD
Processor Specs	Integrated CPU/GPU • 4GB RAM 64 GB Storage Additional memory option via USB-C • Integrated WiFi & Bluetooth
Augmented Reality (AR) or Mixed Reality (MR) capabilities	AR & MR with built-in VisionEye SDK
Camera & Battery	13 Megapixels • 1750 mAh [single rechargeable battery]
ThirdEye App Store	Yes
Unity & Android Studio Integration	Yes [SDK provided for VisionEye SLAM, Audio & Gaze UI]
User Interface	Gaze • Audio • Wireless controller optional • Gestures Optional
Smart Glasses Optics Look	Thin optical waveguide display • Wide peripheral vision • Smallest form factor
Light Transmission	-90% of outside light
Industrial Certifications	Available. Can connect to HardHat for ANSI Z87.1 certification

3.3.2.3 Vuzix Blade Smart Glasses



Figure 20: Blade Smart Glasses [16].

The Vuzix Blade Smart Glasses is a pair of AR glasses made by Vuzix, a manufacturer from the US. Vuzix's blade AR Smart Glasses are suitable for professionals as well as consumers. The Blade's lenses provide standard UV protection and it is possible for users to customize them with prescription inserts. These augmented reality smart glasses have a built-in Android OS. However, these Alexa-enabled AR glasses can also connect to users' devices via Wi-Fi and/or Bluetooth. The below table presents the specifications of the Vuzix Blade Smart Glasses.

Table 8: Vuzix Blade Smart Glasses Specifications [17].

Optics and Electronics	Waveguide based see through optics Vibrant full color display Right eye monocular Quad Core ARM CPU Megapixel camera (with auto-focus) Android OS
Controls	Voice control – multilingual Touch pad with gesture Head motion trackers Haptic vibration alerts Companion app for Android & iOS devices
Audio	Micro USB ear-phone jack Full BT functionality Noise canceling microphones Built-in audio
Battery	Internal LiPo rechargeable batteries
Versatile Eyeglass Options Available	Prescription inserts All lenses standard with UV protection
Connectivity	microSD expansion slot 2.4GHz WiFi and Bluetooth wireless Micro USB

3.3.2.3 *Toshiba dynaEdge AR100 Viewer*



Figure 21: Toshiba dynaEdge AR100 Viewer [18].

The Toshiba dynaEdge AR100 Viewer is an augmented reality head-mounted display made by Toshiba, a Japanese company. The dynaEdge AR100 Viewer is a head-mounted display for professionals working in manufacturing or logistics as well as remote experts where collaboration is a key factor in performance. Furthermore, Toshiba sells the dynaEdge AR100 Viewer without a mounting solution. However, this AR viewer fits over most traditional glasses, consequently turning them into smart glasses. Below is the table of Toshiba dynaEdge AR100 Viewer specifications.

Table 9: Toshiba dynaEdge AR100 Viewer Specifications [18].

Optics	0.26" Diagonal PixelPure™ Display Display Resolution: 640 x 360 Content Resolution: 1280 x 720
Camera	5 Megapixel, F2.8, Focus Range: 10cm to infinity, Auto Focus Video Capture: Up to 1080p @ 30fps Flash LED
Connectors	USB Type-C™
Audio	Built-in Speaker Dual Microphones
Sensors	Proximity Sensor Ambient Light Sensor (ALS) Gyroscope Accelerometer Compass GPS
Versatile Mounting Options Available	AR100 Lens-less Frame AR100 Safety Frame AR100 Safety Helmet Mount AR100 Headband Mount
Weight	47g / 1.65oz.
Navigation Controls	3 Control Buttons Touchpad

In the current configuration, the EPSON Moverio BT-300 Smart glasses seem to present the most desired capabilities and the apt compatibility with the Drone and the respective AR software that is used.

3.4 Trend analysis view

This view of the multi-purpose dashboard allows the user to select a time period and visualize the trend of production of a PV plant. The user can select a timeslot of historical PV production and request the results of the trend analysis algorithm.

Table 10: Trend Analysis view characteristics.

	Description
Functionality description	The navigators of this view can monitor and spot the change of no linear to linear trend change regarding a PV panel array production.
Visualization capabilities	This view shows the intervals of upper and lower limits where a trend change is spotted. The user can choose a timeslot and request the trend analysis output. This is visualized in two graphs, one showing the actual PV generation while the other shows the trend analysis algorithm [4] output.
Technical specifications	<p>This view is developed by utilizing the Angular 8 framework, NodeJS and MySQL-MariaDB database, Python and Flask in order to call an API to the Angular 8 front-end to run the trend analysis python implementation.</p> <p><i>Server Hardware</i> CPU: Intel(R) Core(TM) i5-6600 CPU @ 3.30GHz Memory: 16GB</p> <p><i>Software</i> Operating System: Windows 10 Pro MySQL version: 10.1.19-MariaDB NodeJS version: 12.0.0 Python version: 3.7 Flask version: 1.0.2 Sequelize Framework Version: ^5.8.5</p>
The external dependencies include	There are no external dependencies.
Input data	<p>The input data is a json entry that contains the actual PV generation data from CERTH/ITI nZEB Smart Home.</p> <p>For example, the input data is:</p> <ol style="list-style-type: none"> 1. [{measurementDate: "2020-04-07 23:00:00+00:00", value: 0},...] 1. 0: {measurementDate: "2020-04-07 23:00:00+00:00", value: 0} 2. 1: {measurementDate: "2020-04-07 23:15:00+00:00", value: 0} 3. 2: {measurementDate: "2020-04-07 23:30:00+00:00", value: 0} 4. 3: {measurementDate: "2020-04-07 23:45:00+00:00", value: 0} 5. 4: {measurementDate: "2020-04-08 00:00:00+00:00", value: 0} 6. 5: {measurementDate: "2020-04-08 00:15:00+00:00", value: 0}

	7. ...
Output data	<p>The output data updated the two graphs of the view depending on the user's navigation choices.</p> <p>For example, the output data is:</p> <pre>{analysis: [{measurementDate: "2020-04-08T06:30:00.000Z", value: 0.45713509760418125},...],...}</pre> <ol style="list-style-type: none"> analysis: [{measurementDate: "2020-04-08T06:30:00.000Z", value: 0.45713509760418125},...] forecasted: [{measurementDate: "2020-04-07 23:00:00+00:00", value: 0},...] limits: {t1: 2.048407141795244, t2: 1.3125267775925376} real: [{measurementDate: "2020-04-07 23:00:00+00:00", value: 0},...] sections: [23, 6, 11, 4, 22] step: 66
Implementation features and delivery	<p>Currently the tool is hosted by a web server and is released as an Angular 8 project that contains multiple components.</p> <p>The source code is versioned on the eDREAM gitlab repository.</p>

Some screenshots of the tool are reported.

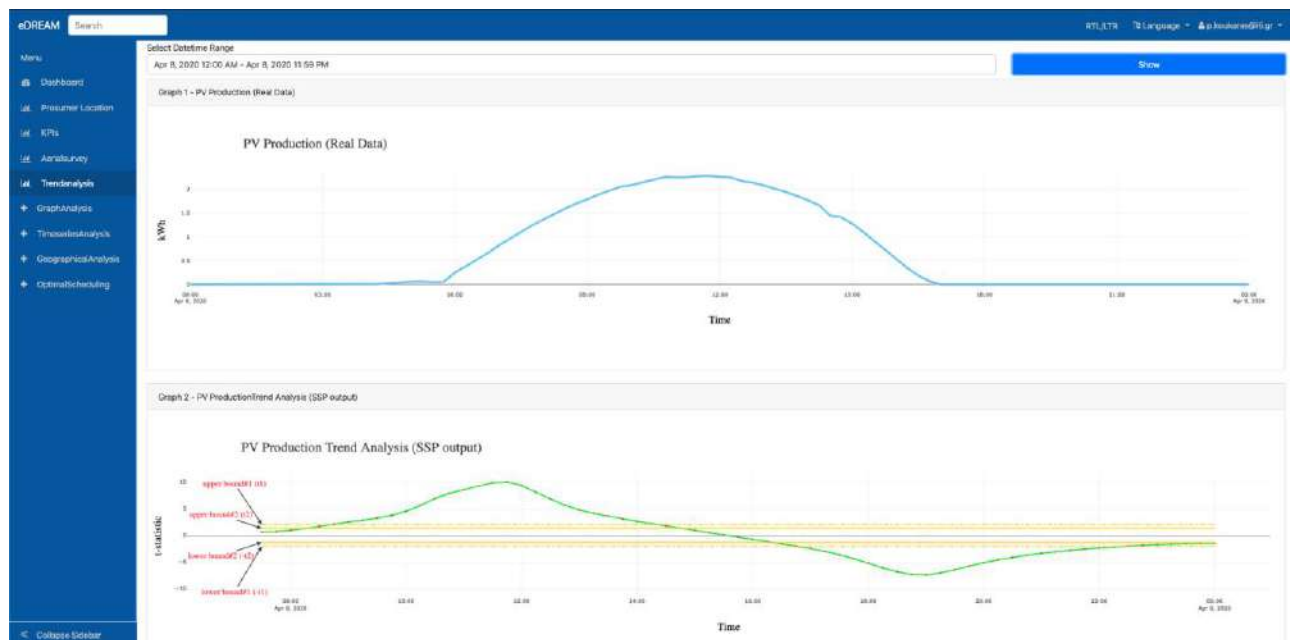


Figure 22: Trend Analysis output for 1 day.



Figure 23: Trend Analysis output for 3 days.



Figure 24: Trend Analysis output for 7 days.

3.5 Multi-objective visual clustering for common behaviors

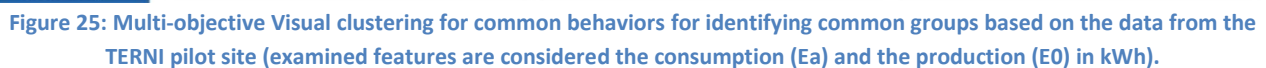
This view of the multi-purpose dashboard allows the user to interact with 3 graphs that form spanning trees related with specific prosumer features, aiming to monitor, visualize and identify common behaviors and habits.

Table 11: Multi-objective Visual clustering view characteristics.

	Description
Functionality description	The navigators of this view can monitor and spot common behaviors of prosumers regarding their energy consumption/production habits.
Visualization capabilities	This view shows the common behaviors of prosumers on a specific timeslot. The user can choose a timeslot, two energy features (Ea (kWh), E0 (kWh), Demand Flexibility (kWh), CO2_reduction (kWh)) and request the graph analysis output. This is visualized in three THD minimum spanning tree graphs, one for each of the selected features and one for the merge of the selected features.
Technical specifications	<p>This view is developed by utilizing the Angular 8 framework, NodeJS and MySql-MariaDB database as well as a custom made for multi-objective optimization (rxcomp.js) and Vega (for force-directed graphs) JavaScript libraries.</p> <p><i>Server Hardware</i> CPU: Intel(R) Core(TM) i5-6600 CPU @ 3.30GHz Memory: 16GB</p> <p><i>Software</i> Operating System: Windows 10 Pro MySql version: 10.1.19-MariaDB NodeJS version: 12.0.0 Vega version: 5.3.4 Rxcomp version:0.1 Sequelize Framework Version: ^5.8.5</p>
The external dependencies include	There are no external dependencies.
Input data	<p>The input data is a json entry that contains the data from the prosumers that are grouped and fed according to specified option to the multi-objective service.</p> <p>For example, the input data is a grouper json object and an optionsmaker json object: [grouper] Array(87)</p> <ol style="list-style-type: none"> 0: {id: "1", Ea_2017-04-27 09:00:00: 27, Ea_2017-04-27 09:15:00: 21, Ea_2017-04-27 09:30:00: 2, Ea_2017-04-27 09:45:00: 12, ...} 1: {id: "2", Ea_2017-04-27 09:00:00: 316, Ea_2017-04-27 09:15:00: 251, Ea_2017-04-27 09:30:00: 274, Ea_2017-04-27 09:45:00: 181, ...}, ... 86: {id: "50", Ea_2017-04-27 09:00:00: 25, Ea_2017-04-27 09:15:00: 28, Ea_2017-04-27 09:30:00: 2, Ea_2017-04-27 09:45:00: 0, ...}

	<p>4. 87: {id: "51", Ea_2017-04-27 09:00:00: 0, Ea_2017-04-27 09:15:00: 0, Ea_2017-04-27 09:30:00: 0, Ea_2017-04-27 09:45:00: 0, ...}</p> <p>[optionsmaker]</p> <p>Object</p> <p>features: Array(2)</p> <p>1. 0:</p> <p>1. attributes: (12) ["Ea_2017-04-27 09:00:00", "Ea_2017-04-27 09:15:00", "Ea_2017-04-27 09:30:00", "Ea_2017-04-27 09:45:00", "Ea_2017-04-27 10:00:00", "Ea_2017-04-27 10:15:00", "Ea_2017-04-27 10:30:00", "Ea_2017-04-27 10:45:00", "Ea_2017-04-27 11:00:00", "Ea_2017-04-27 11:15:00", "Ea_2017-04-27 11:30:00", "Ea_2017-04-27 11:45:00"]</p> <p>2. distFun: "l1"</p> <p>2. 1: {attributes: Array(12), distFun: "l1"}</p>
Output data	<p>The output data updated the two graphs of the view depending on the user's navigation choices.</p> <p>For example, the output data is:</p> <p>[multiobjectiveService]</p> <p>Object</p> <p>1. nodes: (87) [{...}, {...}, ...]</p> <p>2. individual: Array(2)</p> <p>1. 0: (86) [{...}, {...}, {...}]</p> <p>2. 1: (86) [{...}, {...}, {...}]</p> <p>3. merged: Array(166)</p> <p>1. [0 ... 99]</p> <p>2. [100 ... 165]</p> <p>3. length: 166</p> <p>The object attributes individual [0] and individual [3] contain the nodes with the appropriate links and the merged contains the grouped data of the two individual attributes.</p>
Implementation features and delivery	<p>Currently the tool is hosted by a web server and is released as an Angular 8 project that contains multiple components.</p> <p>The source code is versioned on the eDREAM gitlab repository.</p>

Some screenshots of the tool are reported.



3.6 Multi-objective visual clustering with specific feature selection

The view of the multi-purpose dashboard leverages on the same technologies already presented but it allows for complete feature selection of the attributes to be used as input to the multi-objective visual clustering.

Table 12: Multi-objective Visual clustering with specific feature view characteristics.

	Description
Functionality description	The navigators of this view can monitor and spot common behaviors of prosumers regarding their energy consumption/production habits.
Visualization capabilities	This view shows the common behaviors of prosumers bases on all historic data. The user can choose up to 4 features to be clustered (attempting to find and group the most similar features) and request the graph analysis output. This is visualized in a force-directed graph. Also, the user can click on a node and get some basic information about the node's attributes.
Technical specifications	<p>This view is developed by utilizing the Angular 8 framework, NodeJS and MySQL-MariaDB database as well as a custom made for multi-objective optimization (rxcomp.js) and Vega (for force-directed graphs) JavaScript libraries.</p> <p><i>Server Hardware</i> CPU: Intel(R) Core(TM) i5-6600 CPU @ 3.30GHz Memory: 16GB</p> <p><i>Software</i> Operating System: Windows 10 Pro MySQL version: 10.1.19-MariaDB NodeJS version: 12.0.0 Vega version: 5.3.4 Rxcomp version:0.1 Sequelize Framework Version: ^5.8.5</p>
The external dependencies include	There are no external dependencies.
Input data	<p>The input data is a json entry contains the selected features of the user that interacts with the view.</p> <p>For example, the input data is json object with the following attributes for all prosumers:</p> <ol style="list-style-type: none"> 1. Id: "1" 2. Datetime: "2017-04-27 09:00:00" 3. start_coords_x: "42.573195801194025" 4. start_coords_y: "12.647669464222965" 5. Ea_kWh: "27.0" 6. E0_kWh: "0.0" 7. Demand Flexibility_kWh: "70.0" 8. CO2_reduction_gCO2/kWh: "69.92943442982872"

Output data	<p>The output data updates the graph of the view depending on the user's navigation choices.</p> <p>For example, the output data is: [Graph service response]</p> <pre>{nodes: Array(1044), individual: Array(2), merged: Array(2069)}</pre> <ol style="list-style-type: none"> 1. nodes: (1044) [{...}, ...] 2. individual: (2) [Array(1043), Array(1043)] 3. merged: (2069) [{...}, ...] <p>The object attributes individual[0] and individual[3] contain the nodes with the appropriate links and the merged contains the grouped data to form the force directed graph.</p>
Implementation features and delivery	<p>Currently the tool is hosted by a web server and is released as an Angular 8 project that contains multiple components.</p> <p>The source code is versioned on the eDREAM gitlab repository.</p>

In the following paragraph some screenshots of the tools are presented, notably, the Multi-objective optimization [3] has been used to plot a force-directed graph in order to spot the similarities (similar features) of the prosumers that best based on the user input criteria.

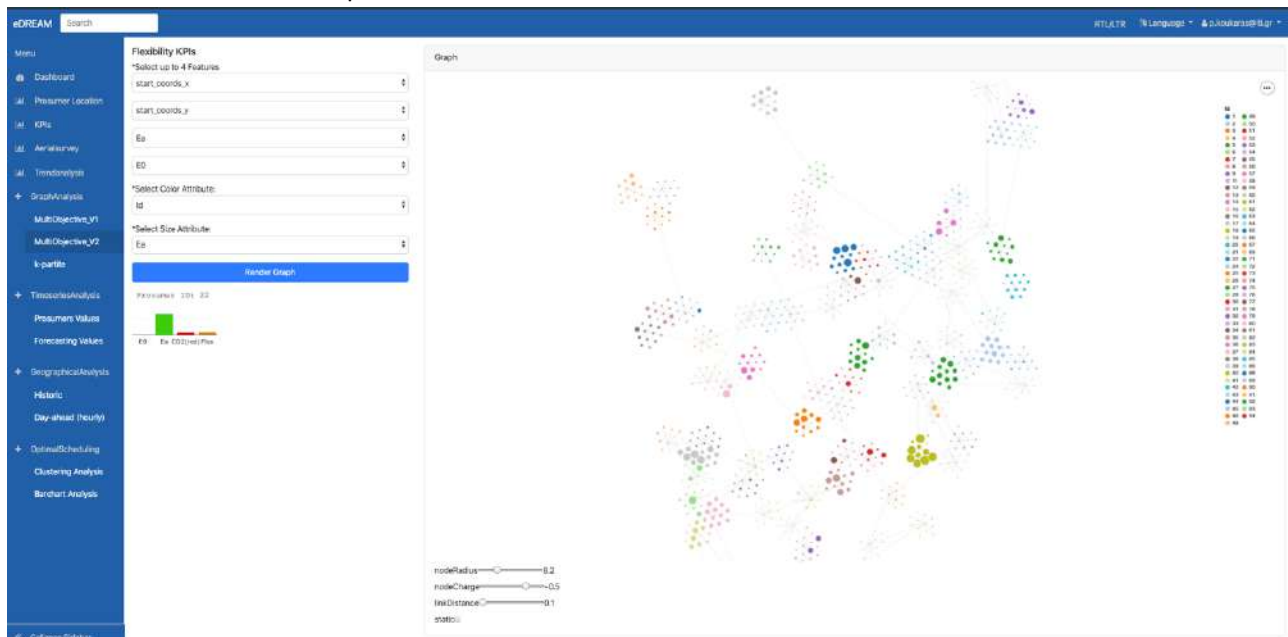


Figure 27: Graph clustering using Multi-objective Optimization for Multimodal Visualization.

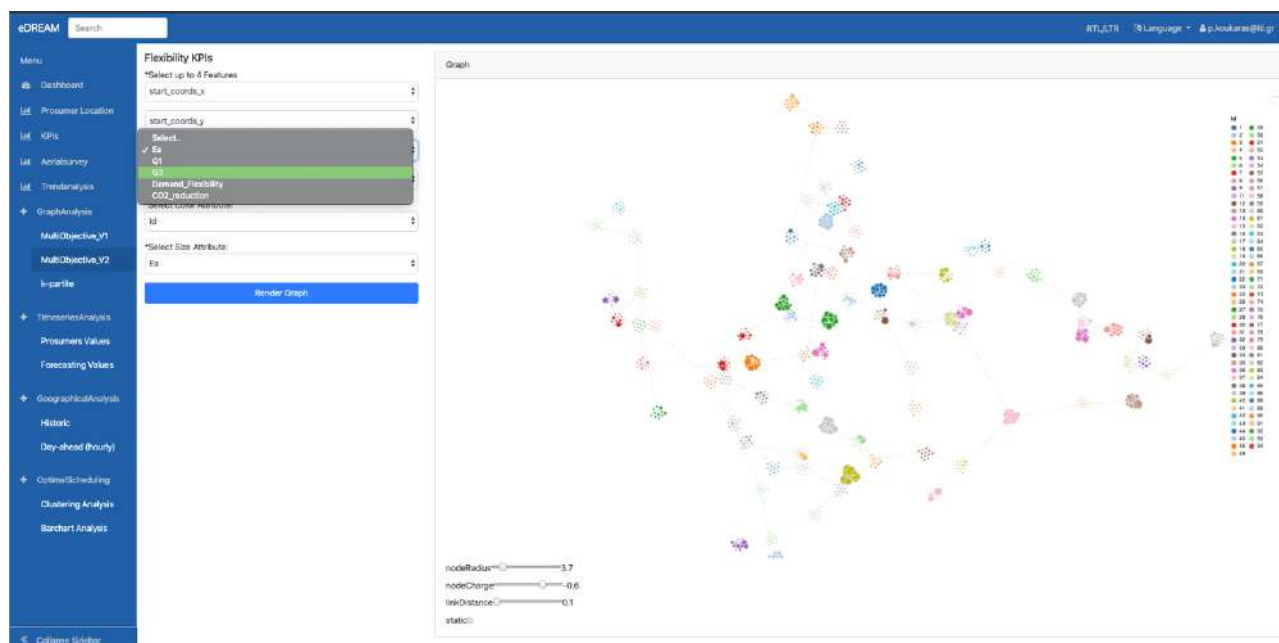


Figure 28: Graph clustering using Multi-objective Optimization for Multimodal Visualization, feature selection.

When clicking on a node on the graph a new graph appears depicting the values of KPIs and the feature attributes that the user selected. This functionality is depicted in the following figure.

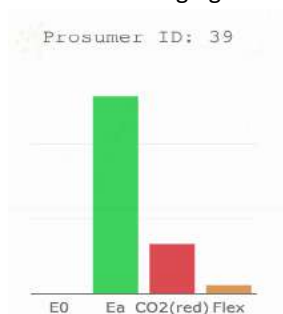


Figure 29: Graph clustering using Multi-objective Optimization for Multimodal Visualization, node feature and KPIs values.

3.7 k-partite visual clustering of common features using binning

This section presents the k-partite visual clustering of common features using binning. On this view the graph represents groups of prosumers that are being binned and plotted utilizing a k-partite graph in a specific time scale based on their levels of consumption (or production respectively or other features). On the current version the binning is performed based on the consumption, in the next version the user will be able to choose the feature based on which the binning will be performed.

Table 13: k-partite visual clustering using binning view characteristics.

	Description
Functionality description	The navigators of this view can monitor and spot common features of prosumers regarding their energy consumption/production.
Visualization capabilities	This view shows the binning of features of prosumers based on a user specified timeslot. This is visualized in k-partite graph.
Technical specifications	<p>This view is developed by utilizing the Angular 8 framework, NodeJS and MySql-MariaDB database as well as a custom made for multi-objective optimization (rxcomp.js) and Vega (for force-directed graphs) JavaScript libraries.</p> <p><i>Server Hardware</i> CPU: Intel(R) Core(TM) i5-6600 CPU @ 3.30GHz Memory: 16GB</p> <p><i>Software</i> Operating System: Windows 10 Pro MySql version: 10.1.19-MariaDB NodeJS version: 12.0.0 Vega version: 5.3.4 Rxcomp version:0.1 Sequelize Framework Version: ^5.8.5</p>
The external dependencies include	There are no external dependencies.
Input data	<p>The input data is a json entry contains the selected features of the user that interacts with the view.</p> <p>For example, the input data is json object with the following attributes for all prosumers picking just the Ea(consumption):</p> <ol style="list-style-type: none"> 1. Id: "1" 2. Datetime: "2017-04-27 09:00:00" 3. start_coords_x: "42.573195801194025" 4. start_coords_y: "12.647669464222965" 5. Ea_kWh: "27.0"

	6. E0_kWh: "0.0" 7. Demand Flexibility_kWh: "70.0" 8. CO2_reduction_gCO2/kWh: "69.92943442982872" Furthermore, a filter function is employed in order to filter the entries based on the user's selection of timeslot. For example: Datetime: "2017-04-27 10:45:00" to Datetime: "2017-04-27 12:45:00".
Output data	The output data updates the graphs of the view depending on the user's navigation choices (timeslot selection). For example, the output data is: <pre>[kpartiteService] {nodes: Array(1114), edges: Array(2088)}</pre> <ol style="list-style-type: none"> nodes: (1114) [{...}, {...},...] edges: (2088) [{...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...},...] The return object contains 2 arrays, the nodes and the edges for the k-partite graph.
Implementation features and delivery	Currently the tool is hosted by a web server and is released as an Angular 8 project that contains multiple components. The source code is versioned on the eDREAM gitlab repository.

Some screenshots of the tool are reported.

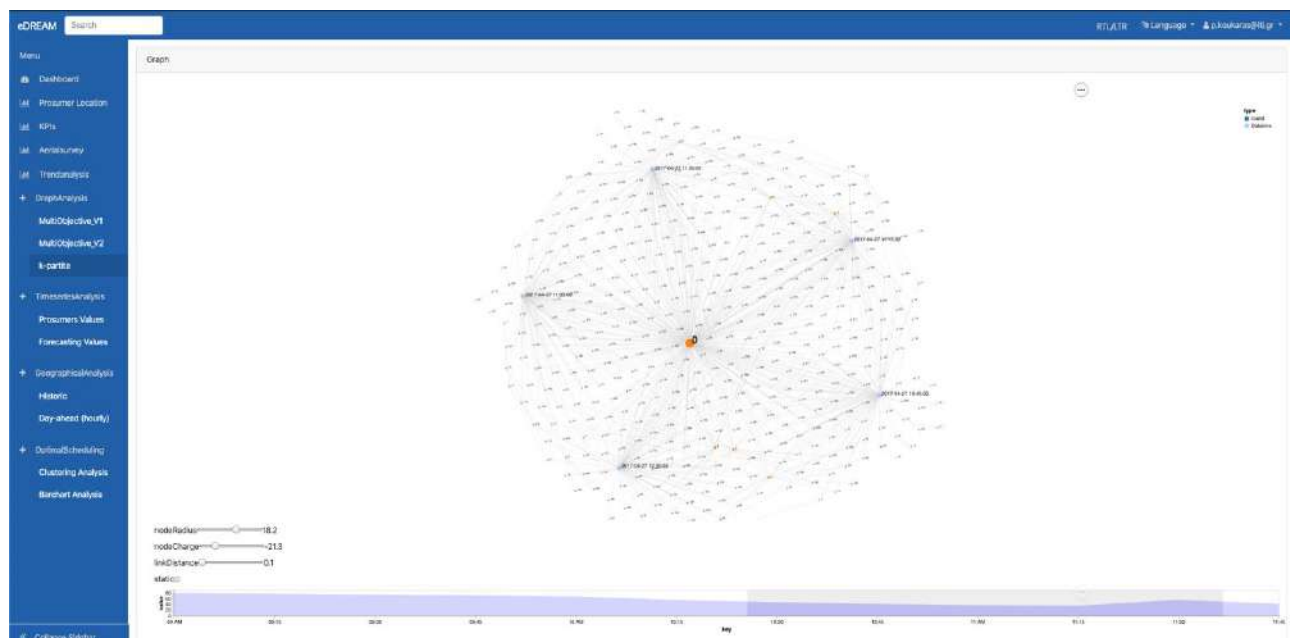


Figure 30: k-partite visualization with binning of prosumer consumptions.

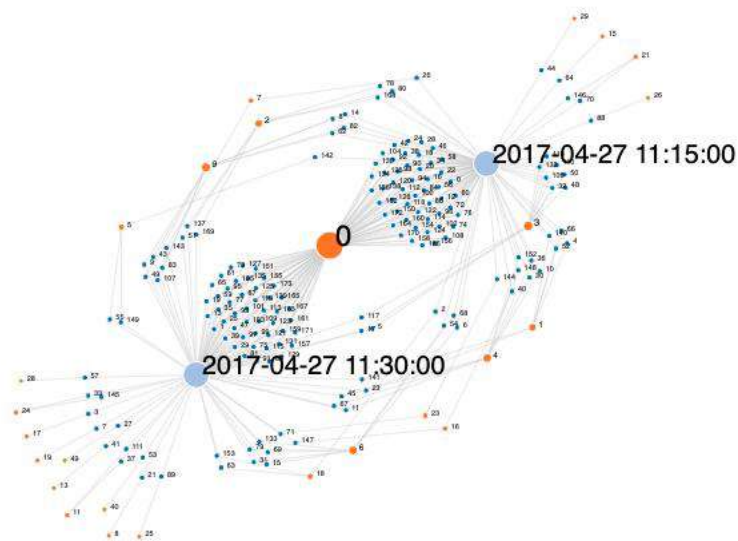


Figure 31: k-partite visualization with binning of prosumer consumptions close look-up.

It should be underlined that changing the features of the k-partite graph, will also affect the resulted layout, and as a result the perceptual formulation of groups, something to be implemented in the future version of this view.

3.8 Geographical analysis utilizing heatmaps

A *Heat Map* is a way of representing the density or intensity value of point data by assigning a color gradient to a raster where the cell color is based on clustering of points or an intensity value. The color gradient usually ranges from cool/cold colors such as hues of blue, to warmer/hot colors such as yellow, orange and hot red. This view describes the spatial distribution of electricity consumption in the area of Terni in Italy. The mapping is based on the intensity of the energy values of each prosumer with an interval of 1-hour.

Table 14: Geographical historic data Analysis using a heatmap view characteristics.

	Description
Functionality description	The navigators of this view can interact with a heatmap in order to monitor the consumptions of all of the prosumers in one of pilot areas (Terni area) according to a user datetime request.
Visualization capabilities	This view shows the density or intensity of aggregated energy consumptions based on the user specified timeslot. This is visualized in a specific area map. While interacting with this map the user is able to zoom in and zoom out of the view dynamically adjusting the heatmap intensity and area coverage. This can be useful in order to spot areas that can be characterized by heavy aggregated consumptions.
Technical specifications	<p>This view is developed by utilizing the Angular 8 framework, NodeJS and MySql-MariaDB database as well as the Mapbox GL JS JavaScript library.</p> <p><i>Server Hardware</i> CPU: Intel(R) Core(TM) i5-6600 CPU @ 3.30GHz Memory: 16GB</p> <p><i>Software</i> Operating System: Windows 10 Pro MySql version: 10.1.19-MariaDB NodeJS version: 12.0.0.1 Mapbox GL JS version: 1.2.0 Sequelize Framework Version: ^5.8.5</p>
The external dependencies include	There are no external dependencies.
Input data	<p>The input data is a json entry contains the selected features of the user that interacts with the view.</p> <p>For example, the input data is json object with the following attributes for all prosumers picking just the Ea(consumption):</p> <ol style="list-style-type: none"> 1. Id: "1" 2. Datetime: "2017-04-27 09:00:00" 3. start_coords_x: "42.573195801194025" 4. start_coords_y: "12.647669464222965" 5. Ea_kWh: "27.0" 6. E0_kWh: "0.0"

	<p>7. Demand Flexibility_kWh: "70.0"</p> <p>8. CO2_reduction_gCO2/kWh: "69.92943442982872"</p> <p>Furthermore, a filter function is employed in order to filter the entries based on the user's selection of timeslot.</p> <p>For example: Datetime: "2017-04-27 10:45:00" to Datetime: "2017-04-27 12:45:00".</p>
Output data	<p>The output data updates the map of the view depending on the user's navigation choices (timeslot selection), as well as updates heatmap's visualization options accordingly. Such options are the following:</p> <ul style="list-style-type: none"> • minOpacity - the minimum opacity the heat will start at. • maxZoom – zoom level where the points reach maximum intensity (as intensity scales with zoom), equals maxZoom of the map by default. • max – maximum point intensity, 1.0 by default. • radius – radius of each “point” of the heatmap, 25 by default. • blur – amount of blur, 15 by default. • gradient – color gradient config, e.g. {0.4: 'blue', 0.65: 'lime', 1: 'red'}.
Implementation features and delivery	<p>Currently the tool is hosted by a web server and is released as an Angular 8 project that contains multiple components.</p> <p>The source code is versioned on the eDREAM gitlab repository.</p>

Some screenshots of the tool are reported.

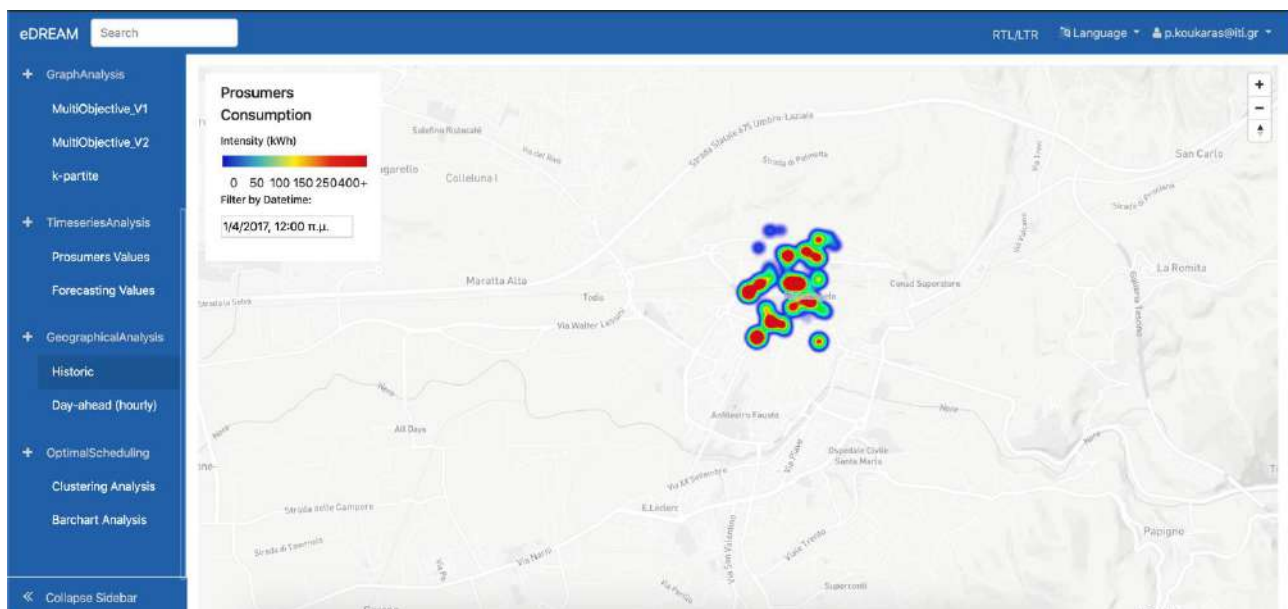


Figure 32: Heatmap of prosumers energy consumption.

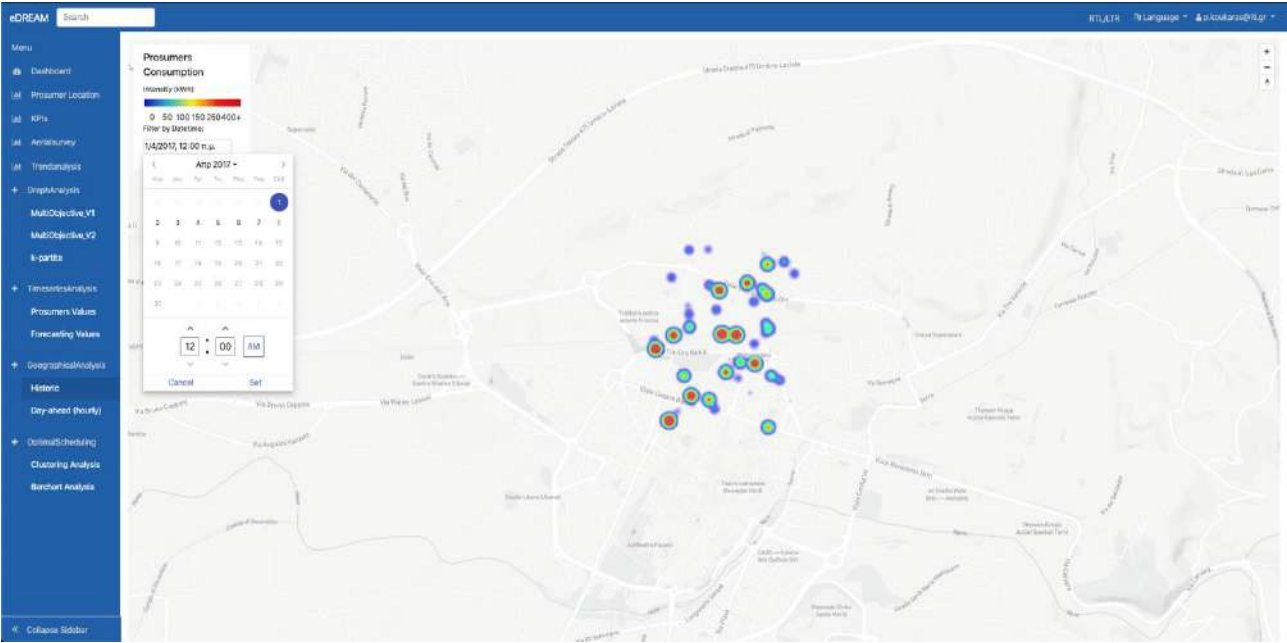


Figure 33: Heatmap of prosumers energy consumption choosing timestamp.



Figure 34: Heatmap of prosumers energy consumption choosing timestamp zoom in and prosumer info.

3.9 Geographical analysis utilizing virtual clusters

The Geographical analysis utilizing virtual clusters interacts with the user by forming virtual clusters of close vicinity aggregated consumptions of multiple prosumers based on their historic data. It merges various functionalities such as zoom in zoom out to the map and thus, allowing the reforming of virtual clusters dynamically.

Table 15: Geographical day-ahead data Analysis with virtual cluster generation view characteristics.

	Description
Functionality description	The navigators of this view can interact with a map in order to monitor the consumptions of all of the prosumers as well as get information about formed virtual clusters regarding consumptions.
Visualization capabilities	This view shows the virtual clusters of aggregated energy consumptions based on the user specified timeslot. This is visualized in a specific area map. While interacting with this map the user is able to zoom in and zoom out of the view dynamically adjusting the clusters' aggregated consumptions. This can be useful in order to spot areas that can form Virtual microgrids or can be characterized by heavy aggregated consumptions.
Technical specifications	<p>This view is developed by utilizing the Angular 8 framework, NodeJS and MySql-MariaDB database as well as the Mapbox GL JS JavaScript library.</p> <p><i>Server Hardware</i> CPU: Intel(R) Core(TM) i5-6600 CPU @ 3.30GHz Memory: 16GB</p> <p><i>Software</i> Operating System: Windows 10 Pro MySql version: 10.1.19-MariaDB NodeJS version: 12.0.0.1 Mapbox GL JS version: 1.2.0 Sequelize Framework Version: ^5.8.5</p>
The external dependencies include	There are no external dependencies.
Input data	<p>The input data is a json entry contains the selected features of the user that interacts with the view.</p> <p>For example, the input data is json object with the following attributes for all prosumers picking just the Ea (consumption):</p> <ol style="list-style-type: none"> 1. Id: "1" 2. Datetime: "2017-04-27 09:00:00" 3. start_coords_x: "42.573195801194025" 4. start_coords_y: "12.647669464222965" 5. Ea_kWh: "27.0" 6. E0_kWh: "0.0" 7. Demand Flexibility_kWh: "70.0"

	<p>8. CO2_reduction_gCO2/kWh: "69.92943442982872"</p> <p>Furthermore, a filter function is employed in order to filter the entries based on the user's selection of timeslot.</p> <p>For example: Datetime: "2017-04-27 10:45:00". This filter will feed to the map view all the consumption of all of the available entries of the prosumers at this timeslot and visualize them.</p>
Output data	The output data updates the map of the view depending on the user's navigation choices (timeslot selection), as well as updates the map visualization options accordingly. The virtual clusters are formed by a function that accumulates the consumption values with close vicinity based on the zoom level that the user specified.
Implementation features and delivery	<p>Currently the tool is hosted by a web server and is released as an Angular 8 project that contains multiple components.</p> <p>The source code is versioned on the eDREAM GitLab repository.</p>

Some screenshots of the tool are reported.

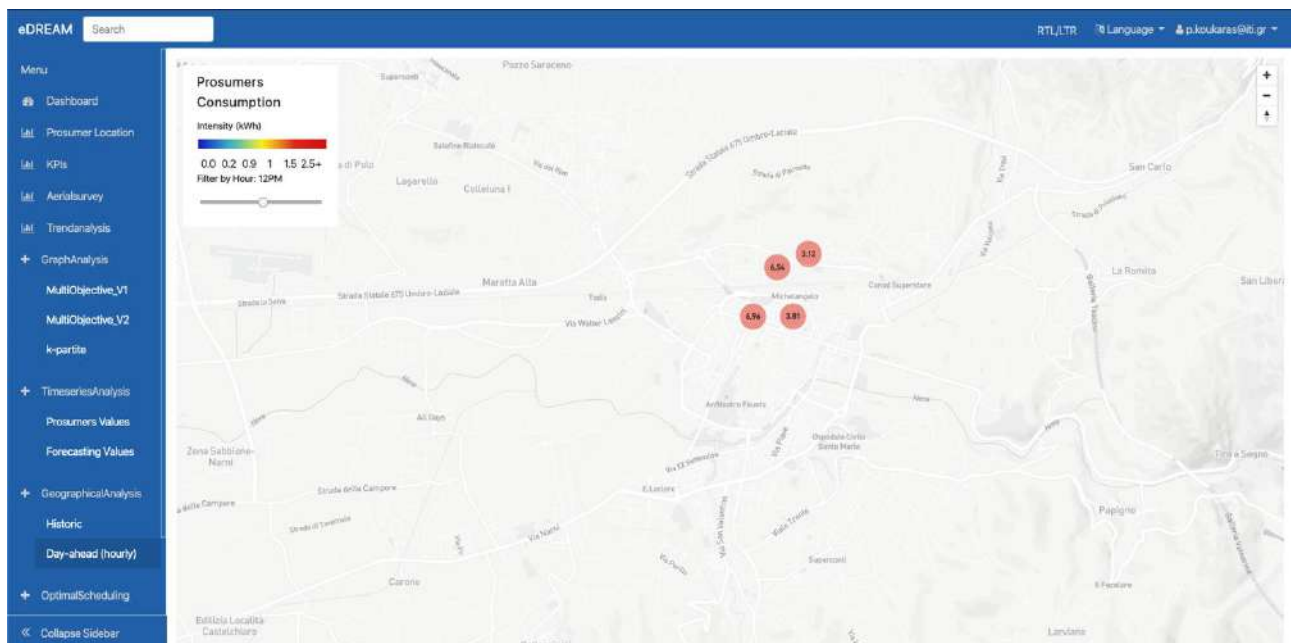


Figure 35: Visualization using clustering for extracting the flexibility for Virtual neighborhoods.



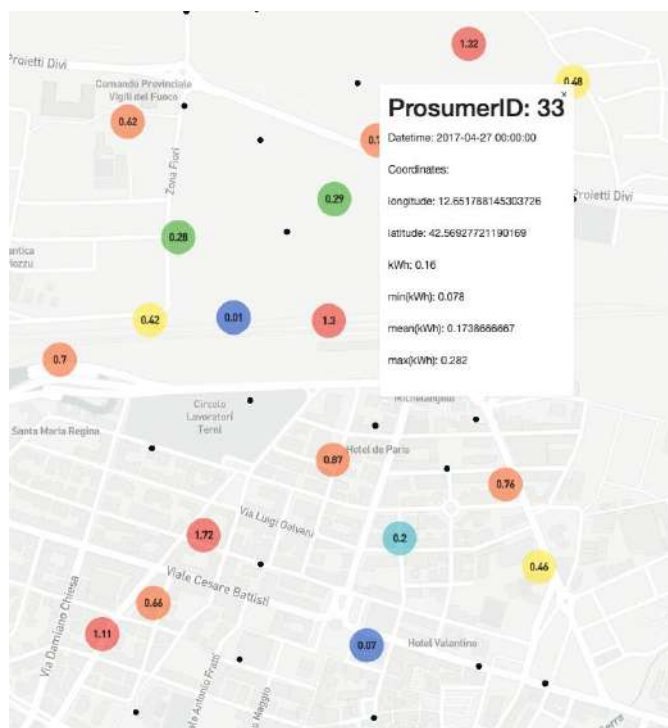


Figure 38: Visualization using clustering for extracting the flexibility for Virtual neighborhoods and info about prosumer.

The above print screens show the interaction of an energy user (Aggregator/DSO) with this view allowing him/her to monitor various areas of Terni regarding their flexibility capacity on user specified timestamps. Zooming in and out on this map, clusters are being dynamically altered in order to represent the close vicinity of users and their combined flexibility.

3.10 Load profiling

On load profiling view, the shown graph illustrates the results of the cluster analysis on the load consumption profiles of the prosumers in a time horizon of 24 hours. Specifically, the thickest curves represent the hourly mean values of the consumption for each one of the clusters, unlike with the rest of the curves that represent individual load consumption profiles.

Table 16: Load Profiling view characteristics.

	Description
Functionality description	The navigators of this view can find out the number of clusters that the prosumers are grouped into, based on the feature of load consumption.
Visualization capabilities	This view shows an intuitive plot of clustering results of the prosumers. For this view we used dummy data.
Technical specifications	<p>This view is developed by utilizing the Angular 8 framework, NodeJS and MySql-MariaDB database as well as the Mapbox GL JS JavaScript library.</p> <p><i>Server Hardware</i> CPU: Intel(R) Core(TM) i5-6600 CPU @ 3.30GHz Memory: 16GB</p> <p><i>Software</i> Operating System: Windows 10 Pro MySql version: 10.1.19-MariaDB NodeJS version: 12.0.0.1 Sequelize Framework Version: ^5.8.5</p>
The external dependencies include	There are no external dependencies.
Input data	<p>The input data is a csv file containing the following entries -we split the day hours in 4 periods (morning, midday, evening, night):</p> <pre>Id,morning,midday,evening,night 0,0.079,0.082,0.101,0.106 1,0.256,0.156,0.289,0.224 2,0.114,0.028,0.145,0.180 3,0.165,0.149,0.273,0.209 4,0.204,0.141,0.218,0.217 ...</pre>
Output data	<p>The output data contain the label of the prosumers in which they were clustered. The output is a csv file containing the following labels as shown below splitting the day hours in 4 periods (morning, midday, evening, night):</p> <pre>Id,label,morning,midday,evening,night 0,0,0.079,0.082,0.101,0.106</pre>

	1,0,0.255,0.156,0.290,0.225 2,0,0.114,0.028,0.145,0.180 3,0,0.165,0.149,0.273,0.209 4,0,0.204,0.141,0.218,0.217
Implementation features and delivery	Currently the tool is hosted by a web server and is released as an Angular 8 project that contains multiple components. The source code is versioned on the eDREAM GitLab repository.

Some screenshots of the tool are reported. The traces on the right on the two figures that follow allow the user to select which prosumers of the portfolio are to be demonstrated on the diagram.

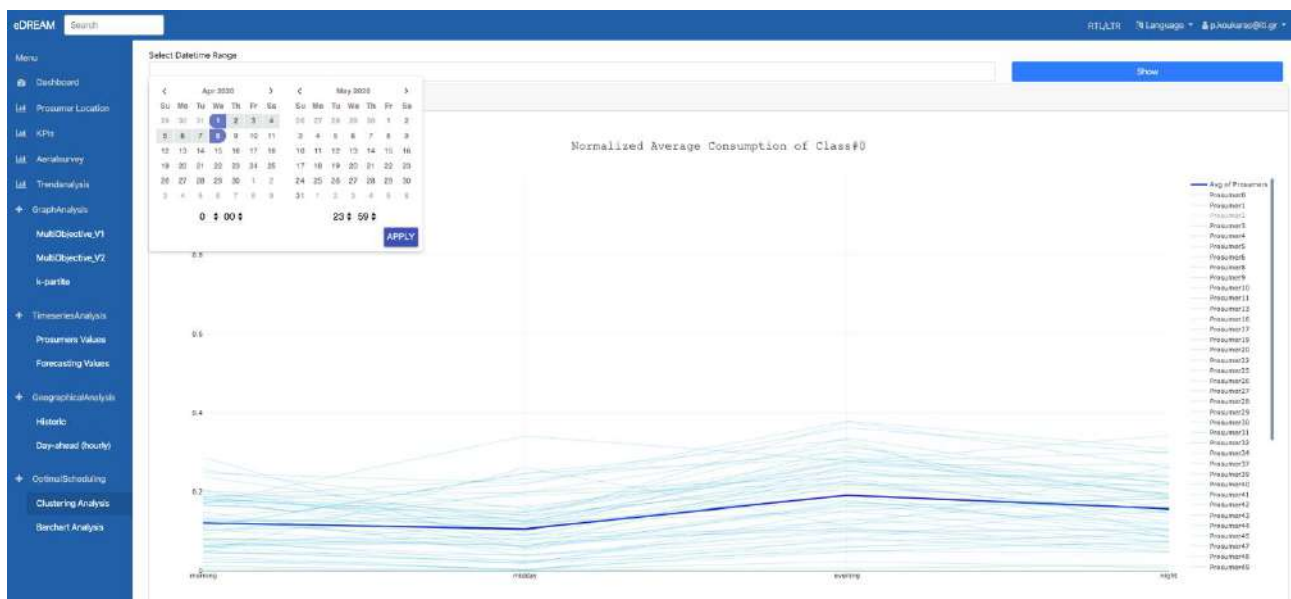


Figure 39: Clustering of prosumers historical load consumption class#0.

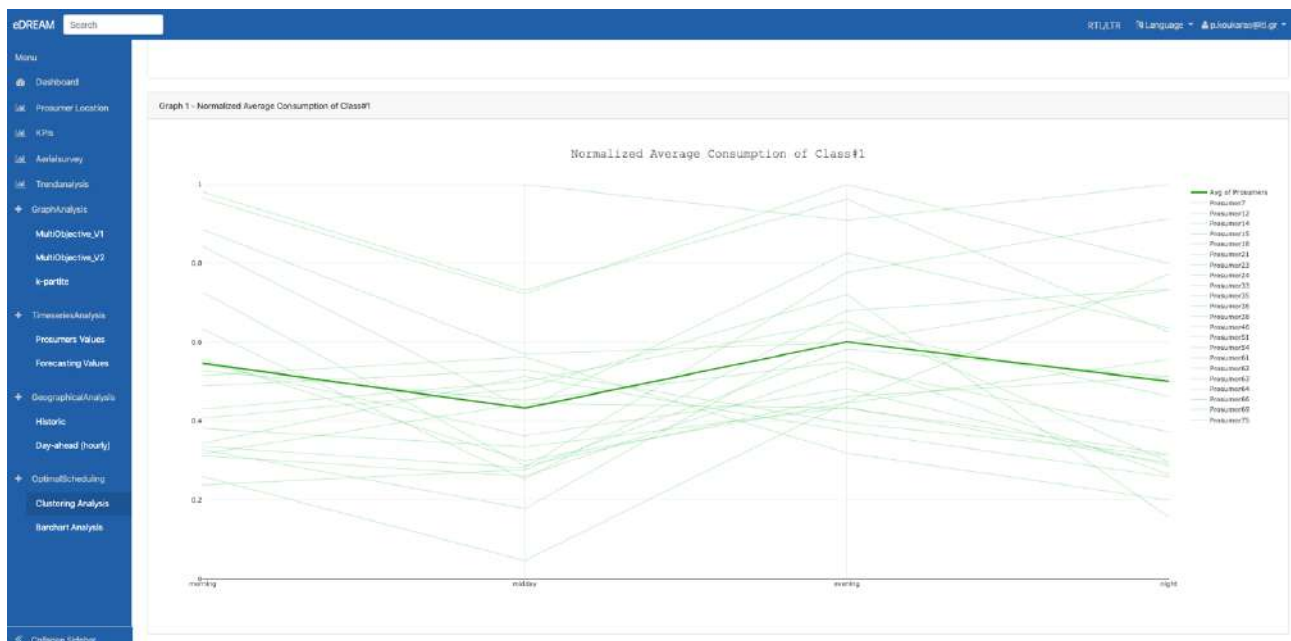


Figure 40: Clustering of prosumers historical load consumption class#1.

3.11 DSS optimal scheduling

On DSS optimal scheduling view the graph illustrates the results of the decision support system regarding the day ahead optimal scheduling of the prosumers.

Table 17: DSS optimal scheduling view characteristics.

	Description
Functionality description	The navigators of this view (e.g. aggregators) can check out the participation rate of prosumer's generation and the load shedding for every time slot to achieve the minimization of the operational cost.
Visualization capabilities	This view shows with different color bars the day ahead scheduling of the assets of an energy aggregator.
Technical specifications	<p>This view is developed by utilizing the Angular 8 framework, NodeJS and MySQL-MariaDB database.</p> <p><i>Server Hardware</i> CPU: Intel(R) Core(TM) i5-6600 CPU @ 3.30GHz Memory: 16GB</p> <p><i>Software</i> Operating System: Windows 10 Pro MySQL version: 10.1.19-MariaDB NodeJS version: 12.0.0.1 Sequelize Framework Version: ^5.8.5</p>
The external dependencies include	There are no external dependencies.
Input data	The input data is the load forecast by estimating the expected demand for every timeslot for the next day. Also, as input data we use the prosumer's maximum capacity of their production and the cost of production as well.
Output data	The output data are 2 csv files, the first one contains which prosumers contribute at each time slot in order to support the demand levels and the second one the load shedding for each time slot.
Implementation features and delivery	<p>Currently the tool is hosted by a web server and is released as an Angular 8 project that contains multiple components.</p> <p>The source code is versioned on the eDREAM GitLab repository.</p>

Some screenshots of the tool are reported. In Figure 41, The day ahead optimal schedule of the prosumers' portfolio is presented, from the aggregator's point of view. The overall demand of the portfolio (bullet line) is shown, along with the generation provided by the prosumers (green bars) and the rest being purchased (orange bars). The latter also demonstrates the demand flexibility margin that needs to be achieved, via shedding, in order to have the maximum reduction of the operation costs. In Figure 42, the aggregators point of view is presented,

where in the top left diagram, the system marginal price for the day ahead is shown (blue bars), along with the forecasted schedule of the aggregated load and generation of the prosumers' portfolio (green and red lines, respectively). In the top right diagram, the pareto front optimal solutions are depicted (red asterisks), among which the aggregator can select one of them, leading each time to a particular optimal resources and consumption scheduling, demonstrated in the two diagrams on the lowest part of the figure.

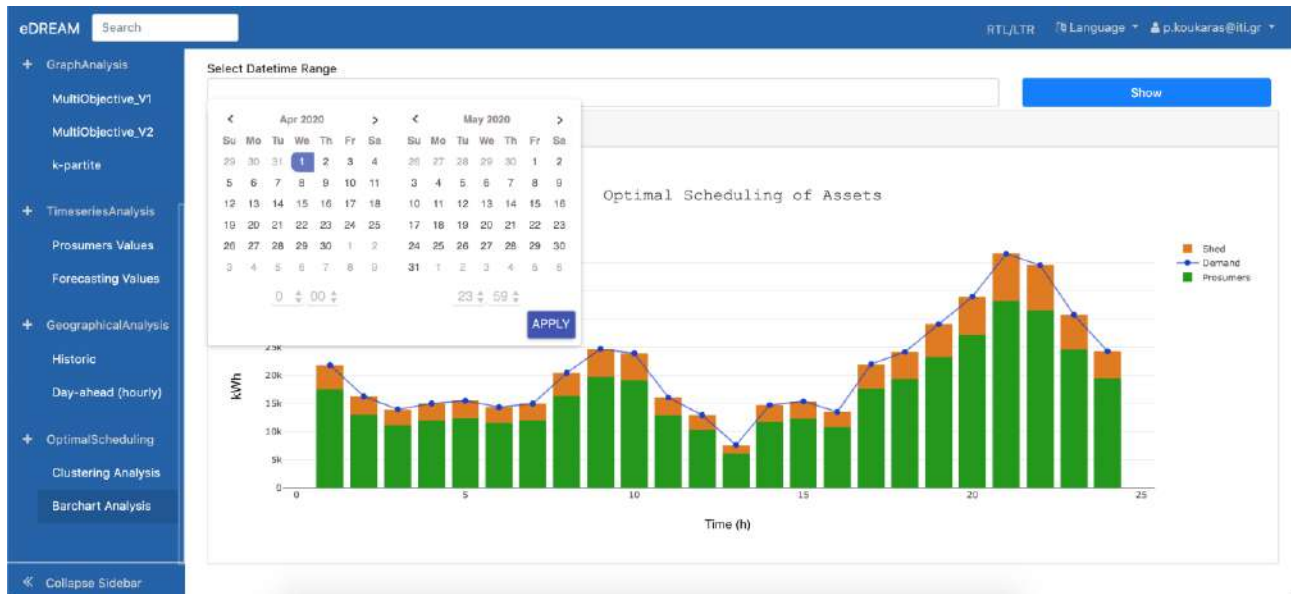


Figure 41: DSS output for day ahead optimal scheduling.

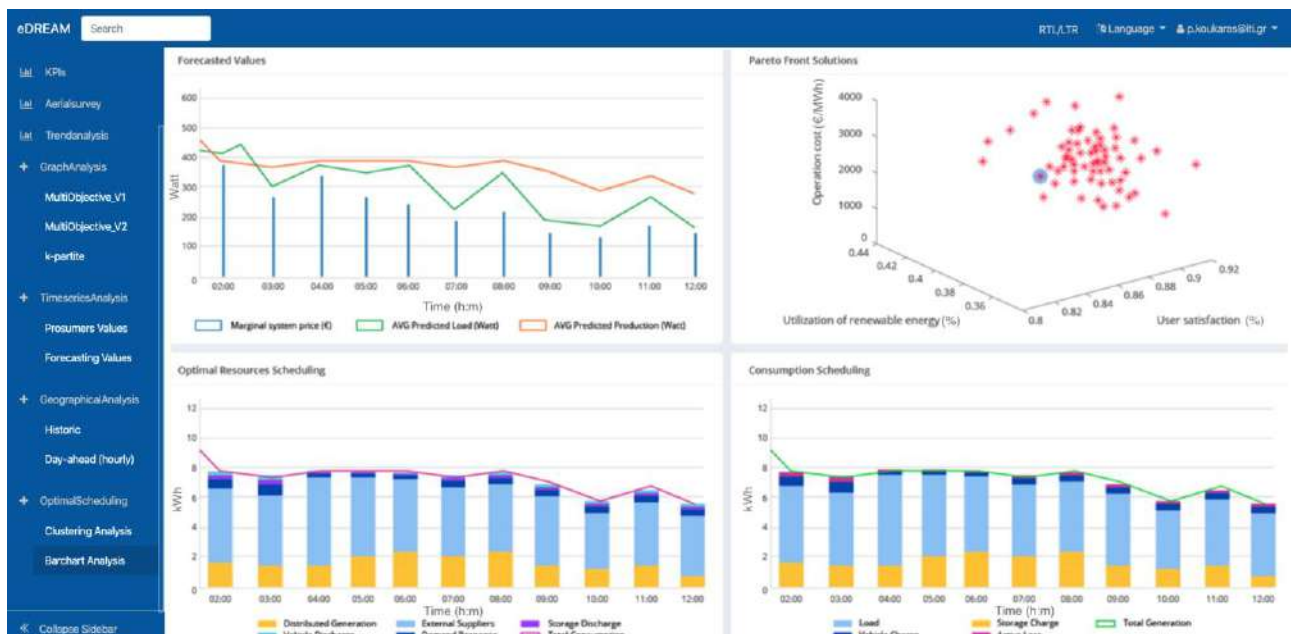


Figure 42: DSS output for optimal scheduling for a particular intraday time period.

4 Integration context

This section is dedicated to the collection and presentation of the summary of the tools that were used for the development of the graph visualization platform. The main goal is to underline the technologies, management criteria, parameters etc. that were utilized during the development of this platform. These can be used as a point

D4.8 – Interactive Visualization framework for improving DR strategies v2

of reference for scientific consideration or development purposes towards Visualization platforms, services and functionalities offered by state-of-the-art energy management systems. In that extend, we refer to the following significant outcomes:

- Features of the operation analysis framework
- Type of I/O data;
- Software/Technologies and Delivery of the tool;

Features of the operation analysis framework

The mostly addressed task features of the Visual analytics techniques seems to be the visualization that utilize maps and multi-objective optimization graph analytics. Out of 11 views offered by the visual analytics platform, 4 utilize maps (Prosumers geolocation, Aerial survey, Geographical Analysis (historic) & Geographical Analysis(Day-Ahead)) and 3 utilize graphs (Multi-objective Visual clustering for common behaviors, Multi-objective Visual clustering with specific feature selection & k-partite visual clustering of common features) while the rest represent a mix of various visualization techniques. This is due to the pilot requirements as stated regarding the eDREAM project.

Type of I/O data

The following type of data have been identified as the most commonly used for in- and outgoing information.

Input data

- Monitoring data
- Visualisation settings
- Load profiles of consumers, prosumers, etc.
- DSO required load profiles
- optimized profiles
- Real-time measurements from energy meters/environmental sensors
- Flexibility Availability
- Energy Prices
- Measurements from grid
- Weather Forecast

Output data

- Power profiles
- Energy consumption
- Forecasted data (consumption, production)
- Available Flexibility of prosumers
- Forecasted Flexibility of prosumers
- Simulated data
- Real-time measurements
- DER status
- Alerts
- Suggested actions
- Notifications

Software/technologies and Delivery of the tools

The programming languages, software packages, or technologies utilized for this deliverable are the following:

- Angular 8

- HTML
- Flask
- NodeJS
- JavaScript
- JQuery
- MySql
- REST
- REST API
- RESTful Services
- Python
- Sequelize Framework
- MariaDB

The tools of this section are delivered as different components of an Angular 8 project in the form of a Docker container and as an embedded executable firmware.

5 Conclusions

This document reports the progress so far regarding the implementation of the eDREAM visualization framework. At first the methodology framework was presented towards the detailed reference of the Visual Analytics as part of the Visualization Framework in accordance with the previous related deliverables. The detailed overview of the visualization techniques presented in this deliverable will continue to be updated and drive the implementation and design of the eDREAM project meeting the pilot requirements. This presentation is performed in a tabular manner comprising 11 different views offered as a unified multi-purpose dashboard for the eDREAM platform users.

The core features and characteristics are being addressed and presented during that implementation phase. In detail, analytics are utilized with clustering and classification scenarios which exploit trends and outliers that feed the output of the Visual analytics platform. That process is supported by various machine learning algorithmic approaches that are incorporated at the back end of the platform. The functionalities are improved since developers are being involved in order to refine and improve the architectural definition of the eDREAM multipurpose dashboard that incorporates the tasks' fulfilment of this deliverable.

As stated in Deliverable 4.4 “Interactive Visualization framework for improving DR strategies V1” this document incorporates the development progress as it was analysed in order to match the expected implementation criteria. Therefore, the aim of this deliverable is to provide the implementation of the high-level description of the tasks that were presented in Deliverable 4.4 in order to produce a prototype of the eDREAM Visual Analytics platform. The development process will continue to be updated until the end of the eDREAM project in order improve the Graph Analytics platform and its functionalities and offer novel and next generation analytics tools for the energy sector.

References

- [1] [eDREAM D4.4] -eDREAM Deliverable D4.4 – Interactive Visualization framework for improving DR strategies V1. April 2019.
- [2] [eDREAM D2.6] - eDREAM Deliverable D2.6 - User group definitions, end-user needs, requirement analysis and deployment guidelines V2. August 2019.
- [3] Kalamaras, I., Drosou, A., & Tzovaras, D. (2014). Multi-objective optimization for multimodal visualization. IEEE Transactions on Multimedia, 16(5), 1460-1472.
- [4] T. Vafeiadis, S. Krinidis, C. Ziogou, D. Ioannidis, S. Voutetakis, D. Tzovaras, "Robust malfunction diagnosis in process industry time series", 14th IEEE International Conference on Industrial Informatics (INDIN'16), Futurescope, Poitiers, France, 18-21 July 2016, pp. 111-116.
- [5] <https://dronekit.io>
- [6] <https://mavlink.io/en/>
- [7] <https://www.dronedeploy.com>
- [8] <https://flytbase.com>
- [9] <https://ardupilot.org>
- [10] <https://www.ugcs.com>
- [11] <https://developer.dji.com>
- [12] <https://www.dji.com/gr/matrice-200-series>
- [13] <https://epson.com/For-Work/Wearables/Smart-Glasses/Moverio-BT-300-Smart-Glasses-%28AR-Developer-Edition%29-/p/V11H756020>
- [14] <https://www.thirdeyegen.com/x2-smart-glasses>
- [15] https://www.dropbox.com/s/5k4uekgvpc6qn9/X2_MRGlasses_Brochure.pdf?dl=0
- [16] <https://www.vuzix.com/products/blade-smart-glasses>
- [17] <https://vuzix-website.s3.amazonaws.com/files/Content/pdfs/vuzix-blade-smart-glasses-d04.pdf>
- [18] <https://us.dynabook.com/smartglasses>