The eDREAM project is co-founded by the EU's Horizon 2020 innovation programme
under grant agreement No 774478

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

DELIVERABLE: D7.2 Technology Verification Report V1

Authors: Paschalis A. Gkaidatzis (CERTH), Christina Tsita (CERTH), Dimosthenis Ioannidis (CERTH)

Imprint

D7.2 Technology Verification Report V1 (Month 28)

Contractual Date of Delivery to the EC:	30.04.2020
Actual Date of Delivery to the EC:	01.05.2020

Author(s):	Paschalis A. Gkaidatzis (CERTH), Christina Tsita (CERTH), Dimosthenis Ioannidis (CERTH)
Participant(s):	Paraskevas Koukaras (CERTH), Napoleon Bezas (CERTH), Giorgios Sfikas (CERTH), Giannis Gotsos (CERTH), Thanos Mekras (CERTH), Victoria Murcia (KIWI), Charalampos Psarros (TU), Fathi Abuchem (TU), Benjamin Hunter (TU), Ugo Stecchi (ATOS), Javier Gomez (ATOS), Maria Lourdes Gallego Miguel (ATOS), Francesco Bellesini (EMOT), Giuseppe Ravenduto (ENG), Tudor Cioara (TUC), Marcel Antal (TUC).

Project:	enabling new Demand Response Advanced, Market oriented and secure technologies, solutions and business models (eDREAM)
Work package:	WP7 – eDREAM validation and reliability analysis
Task:	7.2 – Technology Testing, Verification and Assessment
Confidentiality:	Public
Version:	1.0

Legal Disclaimer

Copyright

Executive Summary

The deliverable D7.2, related to the activities of Task T7.2 entitled Technology Testing, Verification and Assessment, presents a first version of the detailed technical deployment guide providing all the necessary information for the installation, configuration and administration of the eDREAM core architectural components due to be submitted at M28.

The necessary guidelines are provided for the deployment of the eDREAM components in relevant installations in the two pilot sites of the project, ASM and KIWI. The underlying report will be also used during the iterative validation of the concepts and techniques introduced in the eDREAM. The current development and integration status of the tools involved within the project framework are reported along with any software problems that have been encountered, if any, and possible courses of action for their resolution.

As stated in the Description of Action (DoA), the deployment guide will be further updated during the course of the project with the final version of the deployment guide to be provided with its second version, that is Deliverable D7.7 bound for submission in M36 of the project.

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

AC	Alternating Current
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
CBL	Customer Baseline Load
CI/CD	Continuous Integration/Continuous Deployment
CIM	Common Information Model
CPU	Central Processor Unit
DB	Database
DC	Direct Current
DER	Distributed Energy Resource/s
DG	Distributed Generation
DNO	Distribution Network Operator
DoA	Description of Action
DR	Demand Response
DSO	Distribution System Operator
DSS	Decision Support System
eDREAM	enabling new Demand Response Advanced, Market oriented and secure technologies, solutions and business models
EFR	Enhanced Frequency Response
EMS	Energy Management System
EV	Electric Vehicle
FCR	Frequency Control Reserve
FFR	Firm Frequency Response
FTP	File Transfer Protocol

GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
KPI	Key-Performance Indicator
HDD	Hard Disk Drive
HMI	Human-Machine Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IoT	Internet of Things
ISP	Internet Service Provider
LV	Low Voltage
MQTT	Message Queuing Telemetry Transport
NEDC	New European Driving Cycle
NTG	National Transmission Grid
nZEB	Near Zero Energy Buiding
OBD	On-board Diagnostic
OCPP	Open Charge Point Protocol
OVH	On vous héberge (French: We host you)
P2P	Peer-to-Peer
PLC	Programmable Logic Controller
PV	Photovoltaic
RES	Renewable Energy Source/s
REST	Representational State Transfer
RPC	Remote procedure call
RPF	Reverse Power Flow

RTU	Remote Telecoms Unit
SCADA	Supervisory Control and Data Acquisition
SD	Secure Digital
SM	Smart Meters
SMM	Smart Metrology Meter
SMX	Smart Meter Extension
S.O.	Operating System
SSM	Slope Statistic Method
TCP/IP	Transmission Control Protocol/Internet Protocol
TSO	Transmission System Operator
UI	User Interface
UML	Unified Modeling Language
UMTS	Universal Mobile Telecommunications System
USM	Unbudled Smart Meter
VPP	Virtual Power Plant
VPS	Virtual Private Server
WP	Work Package

1 Introduction

1.1 Scope and objectives of the deliverable and relevance in the eDREAM framework

This deliverable presents a deployment guide for the tools involved within the eDREAM project framework. A development and integration status are reported alongside an updated deployment status of the hardware on each of the eDREAM pilots.

Following the activities of WP3, WP4 and WP5, during which the main development of the tools takes place, and WP6, where the integration of the tools is contemplated, WP7 aims at the validation and verification of the tools as developed within the eDREAM project framework.

Moreover, this deliverable draws close ties with WP2 and its related Tasks, since in that particular WP the eDREAM project platform specifications are determined, including Use Cases, Functional requirements, system specifications, etc. [1]. A general overview of the relations of WP7 with the other WPs of the eDREAM project can be viewed in the figure below.

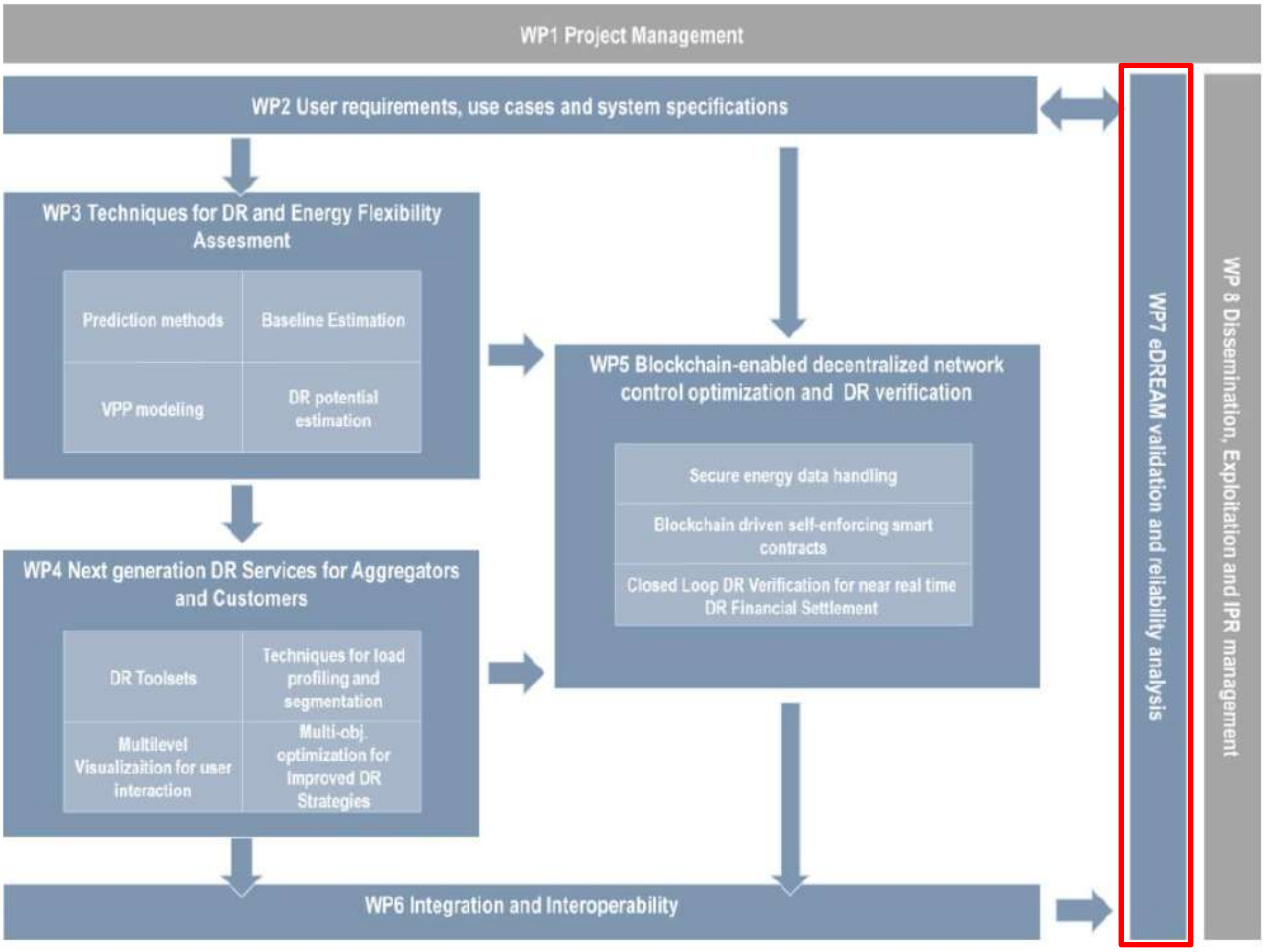

Figure 1. eDREAM PERT diagram showing WP7 (circled in red) relations with other WPs.

To that end, and after following the guidelines as set within the activities of T7.1 [2], where the validation methodology to be utilized in the eDREAM project is being defined, this deliverable presents a first validation report on the tools developed within the eDREAM project framework, since an iterative validation process has been introduced. Therefore, this report is to be considered a living document. A more thorough validation report is to be expected in the second and final version of this document. In the figure below the relations among the various Tasks within WP7 are depicted.

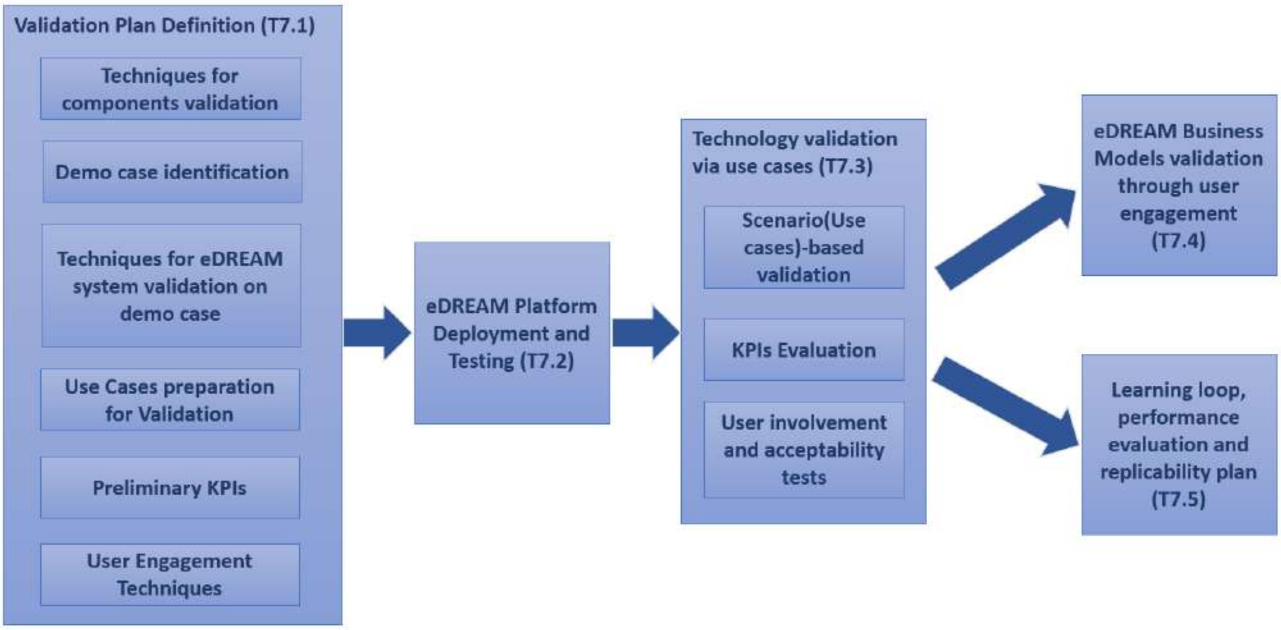

Figure 2. Validation plan definition within the WP7 activities and relations among the WP7 Tasks [2].

1.2 Structure of the deliverable

This deliverable is organized as follows:

- Section 2 presents the implementation status of the architectural components of the eDREAM project framework platform.
- Section 3 presents an updated hardware deployment in the two pilots, that is KIWI and ASM Terni pilots.
- Section 4 is devoted to the conclusions of this deliverable.

2 Implementation Status of Architectural components

This section presents for each eDREAM component, the functional description, how it is connected with other eDREAM components, if connected, the data flow, the development status, the integration status into the overall integrated platform, and finally the problems encountered, if any, during the development and integration of the components. In this section the results of the validation of the individual components against their requirements is presented. Due to the variety of the tools, addressing a vast range of functionalities each tool provider has been responsible for testing its own tools and presenting in this deliverable only the results. It should be stated though that the activities of WP2, where the User requirements, Use Cases and System Specification are determined, WP3, 4 5, and 6, where the main development and integration of the tools are taking place, are still in progress. Therefore, only preliminary information is provided in this deliverable, following the updates of the activities of the aforementioned WPs.

With respect to the tools, the information is based upon the deliverable D2.5 [1] and any updated information provided by the tool provider itself. Thus, there are some tools that, though presented in D2.5, are not presented here, such as the Virtual Power Plant (VPP) and active Microgrid Flexibility Profiling. More information will be provided in deliverable D2.8, D2.9 D2.10 and in the final version of this Deliverable, that is D7.7. Regarding the functional performance test of the tools, several performance test criteria have been used due to the variety of the tools used. Whenever possible the non functional requirements already specified in Deliverable D2.5 [1] have been utilized. Otherwise, each tool provider has provided performance metrics utilized to test the performance of their tools.

The updated validation of those components will be provided in the final version of this deliverable, that is D7.7, along with the overall eDREAM solution validation by its intended users, that is the immediate stakeholders involved in the pilots (KiWi and ASM Terni), which represent the aggregator and Distribution Network Operator (DNO).

2.1 Electricity Consumption/Production Forecasting

2.1.1 Functional Description

The main functionality of this component is to detect prosumer's energy consumption/production patterns and produce accurate predictions of energy supply and demand at different levels of granularities (scale/time). The modules address three different time frames (one hour-ahead, intra-day, day-ahead) and it can manage big data streams of data derived from IoT smart meters. It also provides the capability of combining different prediction techniques through a meta-learner model, in order to extract the most accurate results.
The main functionalities of this components can be summarized as follows:

- Enable creation of various prediction models based on large volumes of historical data;
- Provide predictions results at different granularities (scale/time);

Provide efficient connectivity and communication through RESTful interfaces.

2.1.2 Connection with other Components and Interfaces

The following table presents the necessary information about the implementation and integration stages, so as to achieve adequate connection and communication.

Table 1. Component Connection Requirements.

Component	Connection Type	API Protocol	Data Type	Comments
Input components connections: - *Baseline Flexibility Estimation* Output component connections: - *Virtual Power Plants Generation Modelling and Forecasting;* - *Load Profiling and Disaggregation;* - *PV/RES Degradation & Trend Analysis;* - *VPP & P DR Services s & Optimization Engine;* - *Forecasting Tool* - *DSS &DR Strategies Optimization* - *Blockchain-driven control - LV networks* - *Secured Blockchain driven energy Market*	TCP/IP, HTTP, etc.	RESTful Services	JSON	Receive data from smart meters through a RabbitMQ Broker as well as from other components. Allow querying the prediction results from the output components.

2.1.1 Data Flow

The Electricity Consumption/Production Forecasting module receives historical data of prosumers' consumption/production from the Cassandra DB and the baseline load for each prosumer from the Baseline Flexibility Estimation component. The produced results of consumption/production/flexibility forecasts are available to be retrieved by other components of the platform, such as the Virtual Power Plants Generation Modelling and Forecasting, the Load Profiling, the PV/RES Degradation and Trend Analysis, the VPP & DR Services Optimization engine, the Forecasting Tool, the DSS & DR Strategies Optimization, the Blockchain-driven control for LV networks and the Secured Blockchain-driven energy Market.

2.1.2 Integration Status

The information related to the integration process is mentioned in the following table in a high-level format without many details.

Table 2. Integration process Information.

Integration Status	Under development
Format for Integration	Web service application deployed in a Docker container
Progress up to date	Communication with Cassandra DB for getting monitored data and MySQL DB for saving predictions has been developed
Pending Integration Actions	Interfaces to be tested in real case

2.1.3 Development Status

The below table depicts the relevant information with the development process.

Table 3. Development process information.

Development Status	Under development
Programming Language	Python/JAVA Spring REST
Progress up to date	The main modules of the application, namely the forecasting models for energy consumption/production/flexibility and the ensemble meta-learner have been developed. The modules are able to retrieve data from the database, train basic weak learners, then train the ensemble meta-learner, compute forecast and save the results in the database.
Pending Development Actions	The modules related to the scheduling of the training and forecasting processes to compute automatically the prediction results have to be integrated with the forecasting modules.

2.1.4 Software Problems Encountered

The characteristics of the software problems encountered are summarized in the following table.

Table 4. Software problems characteristics.

Failure Type	Failure Description	Failure Cause	Countermeasure
Time out failure	The system takes a very long time to train the models in case long historical traces are available	High training time for large number of prosumers and long periods of historical data	Create models for clusters of prosumers and train models using only a fixed historical window

2.1.5 Functional performance and efficiency report

A brief description of the results of the verification process, including observations and conclusions from the feedback obtained by the involved actors, is presented in this section. In the case of this tool no performance tests have been conducted yet, since it is under development. More information would be provided in the following version of the deliverable, D7.7.

2.1.6 Recommendations for improvement, future actions and process changes

In the following table recommendations for the improvement of the tool are presented and the adopted future actions taken to enhance the tool, even within the eDREAM project lifetime. This would be further clarified in the next version of the deliverable.

Table 5. Improvement Recommendations.

Recommendations for improvement	Future actions	Description
Improve training time and accuracy by periodically update models when new data is available.	Update the forecasting model parameter re-computation method.	Develop methods to retrain the models using batches of data by periodically update models to decrease training time for long historical traces

2.2 PV/RES Degradation Rate & Trend Analysis

2.2.1 Functional Description

The functionality of this component is to calculate the degradation rate (Rd) at which PV systems modules lose their performance over time. This is a significant information for long term energy production estimation.

The main functionality regarding the Trend Analysis component is to spot the faults from the PV production in order to improve the modeling and get more accurate prediction results. Slope Statistic Method (SSM) is used in order to detect a breakpoint T from no trend to linear trend using a standard parametric linear test, denoted hereafter as t-statistic. Assuming a null hypothesis that the data is stationary, and no linear trend exists, it is concluded that the first candidate for the breakpoint is when the null hypothesis is rejected.

2.2.2 Connection with other Components and Interfaces

The following tables present the necessary information about the implementation and integration stages, so as to achieve adequate connection and communication.

Table 6. Component Connection Requirements PV/RES Degradation

Component	*Connection Type*	*API Protocol*	*Data Type*	*Comments*
Input/output components connections	HTTP	RESTful Services	JSON/ CSV	Receives historical PV production data. Sends the degradation rate and estimation of long-term production of the PV

Table 7. Component Connection Requirements Trend Analysis.

Component	*Connection Type*	*API Protocol*	*Data Type*	*Comments*
Input/output components connections are implemented within the multipurpose visualization dashboard	HTTP(GET/POST)	RESTful Services	JSON	Receive data from the PV array located in CERTH/ITI nZEB Smart Home

2.2.3 Data Flow

The degradation rate module receives historical data of prosumers' PV production from the Cassandra DB. The produced results, degradation rate and long-term estimation of PV production are sent to the DSS & DR Strategies Optimization.

The Trend Analysis module receives historical data of production from the nZEB smart home's API (through a http GET request) stored to the Cassandra DB of the multipurpose visualization dashboard, as well as weather data from weather APIs. The produced results are sent to the PV/RES Degradation and Trend Analysis of the platform.

2.2.4 Integration Status

The information related to the integration process is mentioned in the following tables in a high-level format without many details.

Table 8. Integration process Information PV/RES Degradation Rate.

Integration Status	Under development/Final
Format for Integration	Windows application
Progress up to date	Initial Testing
Pending Integration Actions	Communication with Cassandra DB

Table 9. Integration process Information Trend Analysis.

Integration Status	Final version
Format for Integration	Web service application
Progress up to date	Communication with Cassandra DB and Weather APIs has been developed.
Pending Integration Actions	Interfaces to be tested in real case by testing the deployment version on the pilots.

2.2.5 Development Status

The tables below depict the relevant information with the development process.

Table 10. Development process information PV/RES Degradation Rate.

Development Status	Final
Programming Language	C++
Progress up to date	The main standalone functionalities of the component have been implemented
Pending Development Actions	-

Table 11. Development process information Trend Analysis.

Development Status	Final
Programming Language	Python
Progress up to date	The main standalone functionalities of the component have been implemented
Pending Development Actions	-

2.2.6 Software Problems Encountered

The characteristics of the software problems encountered are summarized in the following table. With respect to the PV/RES Degradation Rate compoment, no problems have been reported so far. In the case of Trend analysis component, a slow response time has been detected and contemplated.

Table 12. Software Problems characteristics Trend Analysis.

Failure Type	Failure Description	Failure Cause	Countermeasure
Slow Response Time when requesting the trend analysis output for a very long time period (e.g. more than 7 days)	The system takes too long to return the graphs that represent the trend analysis output	Large volume of historical data or data that do not exist in the database	Run the trend analysis algorithm periodically (e.g. every 5-10 days) and store the output to the database. That way the response time is reduced significantly

2.2.7 Functional performance and efficiency report

A brief description of the verification process, including observations and conclusions from the feedback obtained by the involved actors. With respect to the PV/RES Degradation Rate component no performance tests have been concluded yet so far. With respect to the Trend Analysis component some early tests have been conducted. The results are shown in the following table.

Table 13. Performance test results Trend Analysis.

Performance Requirement	Description	Remarks (Positive/Neutral/Negative)
Average response time	The time that is required to get the visual output since the user request	The response time varies between 2-5 seconds depending on the request time period to be fed to the trend analysis algorithm. For input time periods from 1 day to 7 days the average response time is 4 secs. Neutral
Error rates	**HTTP Error %:** Number of web requests that ended in an error. We performed 100 requests varying from 1 to 3 days of historic data from the nZEB platform data resulting in 2 HTTP error throws. **Result** 2% HTTP error. **Logged Exceptions:** Number of unhandled and logged errors from your application. After running 100 requests there were no unhandled and logged errors.	Positive

	Thrown Exceptions: Number of all exceptions that have been thrown. After running 100 requests there was no thrown exceptions.	
System Availability	The system ensures availability of services 99%, or superior at any time, 24h/24h, 7d/7d, independently from performances of other applications.	Positive
User friendliness	The User Interface shall have a user-friendly look, fully customized to the needs of different stakeholders	Positive
Map visualization	The User Interface shall provide a user interface offering maps Visualization for a more concrete analysis	Positive
Ease of use case#1	The User Interface shall be able to allow an easy discoverability of the actions available	Positive
Ease of use case#2	The User Interface shall be tailored to the end user needs	Positive
Ease of use case#3	The messages provided by the system must be clear and easy to understand	The trend analysis view does not return any messages to the user rather than the output graph visualization Negative
Ease of use case#4	The User Interface must be simple and intuitive	Positive

2.2.8 Recommendations for improvement, future actions and process changes

In the following table recommendations for the improvement of the tool are presented and the adopted future actions taken to enhance the tool, even within the eDREAM project lifetime. Regarding the PV/RES degradation component no recommendations have been reported at this stage. Regarding the Trend analysis component an improvement of the response time is contemplated. This information would be updated in the next version of the deliverable.

Table 14. Improvement Recommendations Trend Analysis.

Recommendations for improvement	Future actions	Description
Improve response times regarding the output of the trend analysis algorithm	Update the data storing process	A weekly historic data storage frequency can be set, so that the database of the trend analysis component can have prestored the requested data

2.3 Baseline Flexibility Estimation

2.3.1 Functional Description

In this project, the most commonly used methods to generate baselines were examined. The test sites were London (UK) and Terni (Italy). The tests provide insights into how, within a European framework, these methodologies behave and the performance comparison between them. This study has resulted in a tool to generate baselines with the different methods studied, and a comparative analysis of the generated baselines.

Comparison of all methods showed that their accuracy levels vary depending on the user, of the time of day, and the consumption pattern. In general terms, the methodology with the best accuracy was that of comparable day with adjustment. The tool developed as part of this research allows to generate baselines with any of the studied methodologies.

This tool is designed to generate a customer baseline load (CBL). The generation process is based on forecasting algorithms which use the electricity consumption information from the past weeks, and the historical weather (temperature). The tool applies different methods with different algorithms, as according to this study some methods are more effective than others in different situations. The methodology applied in this study, with low overall error is the comparable day with multiplication adjustment. The user must choose the most appropriate methodology to generate the baseline.

2.3.2 Connection with other Components and Interfaces

This component does not interface directly with any other components.

Table 15. Component Connection Requirements.

Component	*Connection Type*	*API Protocol*	*Data Type*	*Comments*
Input components connections: - Cassandra DB	N/A	N/A	CSV	The developed software does not communicate with other software it takes the information directly from existing files with .csv extension

2.3.3 Data Flow

The tool asks the user to specify a date and a file, with the historical electricity consumption data and the weather data (the files need be in .csv format). The user then has to select from a list, which one CBL methodology will be applied. Finally, the tool generates a baseline.

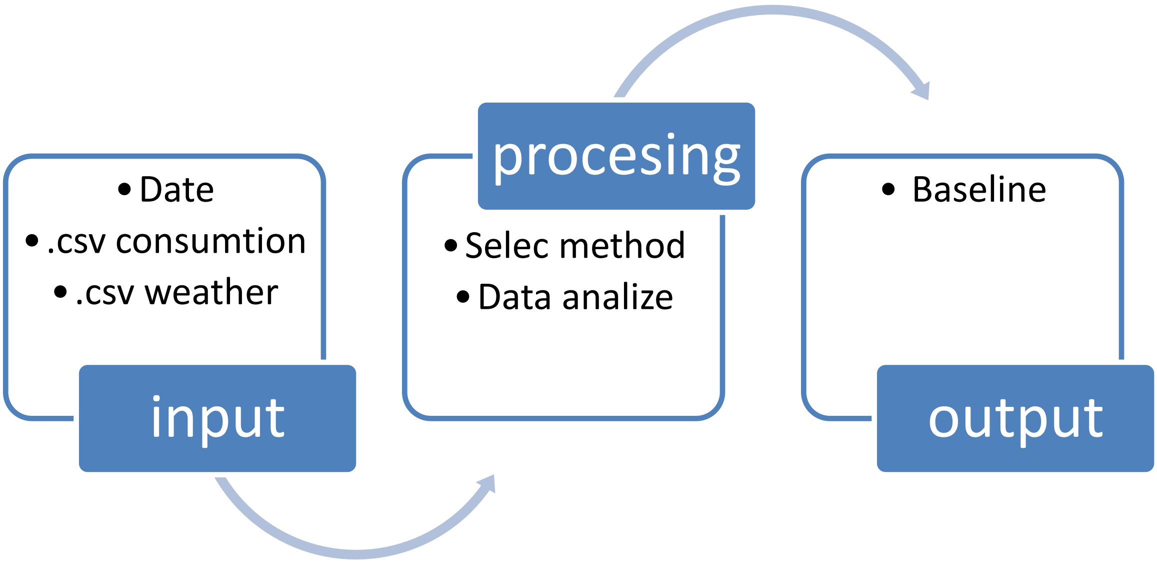

Figure 3. Data flow diagram for Baseline Flexibility Estimation Tool.

To generate the baseline, the designed tool uses a .csv file with the data of electrical consumption in steps of 30 minutes. The data structure must be as follows: 'year, month, day, hour, kW', as depicted in the figure below.

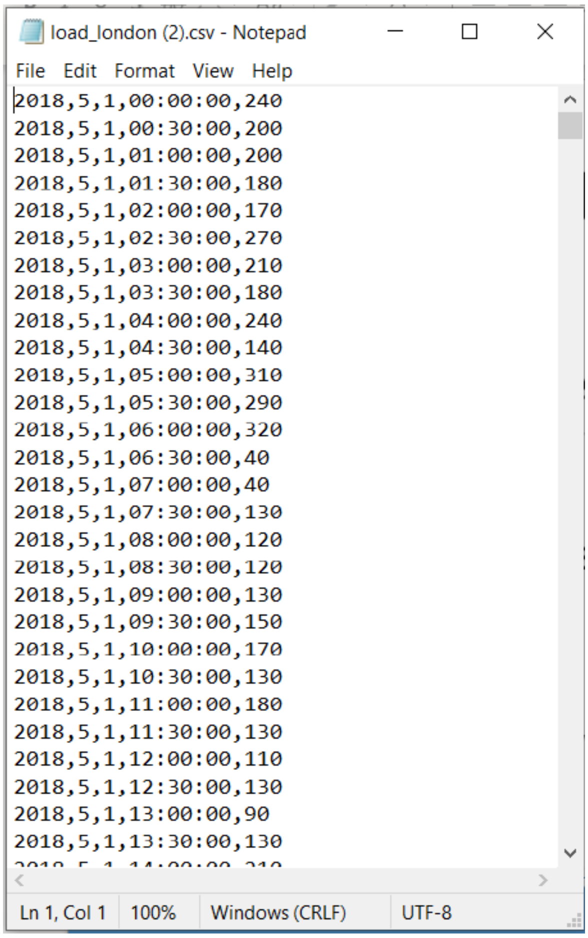

Figure 4. Structure of the electrical consumption data in the .csv file.

In eDREAM project different forecasting algorithms were described and tested. This tool uses algorithms to generate the baselines. In this case, the user can choose which algorithm is most suited to the given task. All algorithms and methods are described in Deliverables 3.2 [3], already delivered, and 3.6 [4], to be delivered in May 2020.

2.3.4 Integration Status

The information related to integration process are mentioned in the following table in a high-level format without many details.

Table 16. Integration process Information.

Integration Status	Not currently integrated
Format for Integration	Not specified
Progress up to date	Not currently integrated with other components
Pending Integration Actions	Integrate with other components

2.3.5 Development Status

The below table depicts the relevant information with the development process.

Table 17. Development process information.

Development Status	Completed
Programming Language	Python
Progress up to date	The tool is in final state
Pending Development Actions	-

2.3.6 Software Problems Encountered

The characteristics of the software problems encountered are summarized in the following table.

Table 18. Software Problems characteristics

Failure Type	Failure Description	Failure Cause	Countermeasure
Interface problems	The interface is not user-friendly	Design	Make improvements to the graphical interface
Data input	Input data is not validated	Design	Make a proper assessment of the data, to ensure that they are coherent

2.3.7 Functional performance and efficiency report

A brief description of the results of the verification process, including observations and conclusions from the feedback obtained by the involved actors, is presented in this section. In so far regarding this particular tool, the tool provider, in this case ATOS, has changed the development of the tool, that is another development environment has been selected along with the input/output formats, in order to make it more homogeneous and compatible with the other tools of the eDREAM platform. Thus, though in final state, no performance tests have been conducted as yet. More information will be presented in the next version of this deliverable.

2.3.8 Recommendations for improvement, future actions and process changes

In the following table recommendations for the improvement of the tool are presented and the adopted future actions taken to enhance the tool, even within the eDREAM project lifetime. This would be further clarified in the next version of the deliverable.

Table 19. Improvement Recommendations.

Recommendations for improvement	Future actions	Description
Classify customers according to their consumption patterns to determine which methodologies are	Use the information collected in this study to create a system that allows customers to be classified	Use a tool to classify customers according to what is the best method for each group. The classification can be done

most accurate for each customer	according to their consumption patterns	using neural networks or linear regression
Carry out more studies in other places in Europe to have a more general vision of the European electrical grid	Study buildings in countries like Spain or Greece	The European electrical network is complex due to the large number of parts that form it. In this study, a European framework was given. However, many of these countries are doing studies to ensure that the tool is applicable to energy customers throughout Europe.

2.4 Virtual Power Plants Generation Modeling & Forecasting

2.4.1 Functional Description

The main functionality of this component is to provide and evaluate energy co-generation optimization algorithms regarding the creation of optimal coalitions, Virtual Power Plants, with the aim of providing more reliable aggregated power sources for providing four services: energy trading service, capacity bidding/selling service, frequency regulation service, and demand response service. The component will select the optimal coalition from a set of enrolled prosumer portofolio, that grouped together have the best offer for the service requested.
The main functionalities of this components can be summarized as follows:

- Enable creation of Virtual Power Plants composed of energy prosumers based on one of the four types of serives implemented: energy trading service, capacity bidding/selling service, frequency regulation service, and demand response service.
- Provide efficient connectivity and communication through RESTful interfaces.

2.4.2 Connection with other Components and Interfaces

The following table presents the necessary information about the implementation and integration stages, so as to achieve adequate connection and communication.

Table 20. Component Connection Requirements.

Component	***Connection Type***	***API Protocol***	***Data Type***	***Comments***
Input components connections: - *Electricity Production/Consumption Forecasting;* - *PV/RES Degradation & Trend Analysis;* - *Cassandra DB* - *Big Data Clustering at Multiple Scale;*	TCP/IP, HTTP, etc.	RESTful Services	JSON	Receive forecast and analysis profiles from input components, in order to compute the optimal coalition, given a profile goal. Save coalition formation results to be accesible by the output components.

Input components connections: - *VPP and active micro-grid flexibility;* - *VPP & DR Services DR services & Optimization Engine;*				

2.4.3 Data Flow

The Virtual Power Plants Generation and Forecasting tool receives as inputs the prosumers energy consumption/production/flexibility forecasts from the Electricity Production/Consumption Forecasting module, the PV/RES degradation analysis from the PV/RES Degradation & Trend Analysis components, grid related information from the Cassandra DB component and information about prosumer consumption/production similarity from the Big Data Clustering at Multiple Scale component. The best coalition according to the service selected and portofolio of available prosumers is saved in the database and will be further used by the VPP and active micro-grid flexibility components and the VPP & DR Services DR services & Optimization Engine.

2.4.4 Integration Status

The information related to the integration process is mentioned in the following table in a high-level format without many details.

Table 21. Integration process Information.

Integration Status	Under development
Format for Integration	Web service application deployed in a Docker container
Progress up to date	Communication with DB for retrieving prosumer consumption/production/flexibility forecasts, coalition computation for the four proposed services and saving the results in the database.
Pending Integration Actions	Interfaces to be tested in real case

2.4.5 Development Status

The below table depicts the relevant information with the development process.

Table 22. Development process information.

Development Status	Under development
Programming Language	JAVA, Spring REST
Progress up to date	The main modules of the application are finalized. Prosumer related information is extracted from the database to generate the prosumer portfolio, one of the four service types can be selected, and the best coalition is computed and saved in the database.
Pending Development Actions	The optimization algorithm used to compute the best prosumer coalition for the selected service type has to be fine-tuned.

2.4.6 Software Problems Encountered

The characteristics of the software problems encountered are summarized in the following table.

Table 23. Software Problems characteristics

Failure Type	Failure Description	Failure Cause	Countermeasure
Out of Memory Error	An out-of-memory error is thrown by the solver used to determine the coalition forming the VPP.	The error appears if the portfolio of prosumers is very large and the optimization algorithm is set to high accuracy, thus requiring a large memory space.	Set a limit to adjust the accuracy of the algorithm based on the portfolio size, thus automatically imposing a memory allocation upper limit.

2.4.7 Functional performance and efficiency report

A brief description of the results of the verification process, including observations and conclusions from the feedback obtained by the involved actors, is presented in this section. In the case of this tool no performance tests have been conducted yet, since it is under development. More information would be provided in the following version of the deliverable, D7.7.

2.4.8 Recommendations for improvement, future actions and process changes

In the following table recommendations for the improvement of the tool are presented and the adopted future actions taken to enhance the tool, even within the eDREAM project lifetime. This would be further clarified in the next version of this deliverable.

Table 24. Improvement Recommendations.

Recommendations for improvement	Future actions	Description
Develop hierarchical coalition formation for all four services.	Implement strategies that allow VPPs to be formed not just from basic prosumers, but also from other smaller VPPs.	Currently, the capacity service allows a Virtual Power Plant to be composed not just from basic energy prosumers, but also from other Virtual Power Plants. The functionality should be extended to the other three services developed.

2.5 Multi-building DR characterization through thermal, optical and LIDAR information fusion

2.5.1 Functional Description

This tools is developed in order to estimate the demand response (DR) of a building having been surveyed and its assets having been identified and its overall energy performance determined, via optical and thermal images. More technical specifications are to be found in deliverable D3.4 [5].

2.5.2 Connection with other Components and Interfaces

The following table presents the necessary information of the implementation and integration stages, so as to achieve adequate connection and communication.

Table 25. Component Connection Requirements.

Component	***Connection Type***	***API Protocol***	***Data Type***	***Comments***
Input components connections: - *DR Aerial Survey Toolkit,* - *Cassandra DB,* - *Baseline Flexibility Estimation* Output component connections: - *Cassandra DB*	TCP/IP, HTTP	Restful Services	JSON	-

2.5.3 Data Flow

After the data are rretrieved from the input components, as reported in the previous Table, they are assessed and the demand flexibility potential and overall energy performance of the building is then stored in the Cassandra DB, in order to be retrieved by the visualization platform, to be presented in the end-user.

2.5.4 Integration Status

The information related to integration process are mentioned in the following table in a high-level format without many details.

Table 26. Integration process Information.

Integration Status	Under development
Format for Integration	Windows application/Web service application
Progress up to date	Communication with the other tools and services
Pending Integration Actions	The whole component to be tested in real cases

2.5.5 Development Status

The below table depicts the relevant information with the development process.

Table 27. Development process information.

Development Status	Under development
Programming Language	Python
Progress up to date	The main standalone functionalities of the component have been implemented
Pending Development Actions	The functionalities related to other components have to be finalized

2.5.6 Software Problems Encountered

The characteristics of the software problems encountered are summarized in this section. With respect ot this tool, no software problems have been reported so far.

2.5.7 Functional performance and efficiency report

A brief description of the results of the verification process, including observations and conclusions from the feedback obtained by the involved actors, is presented in this section. In the case of this tool no performance tests have been conducted yet, since it is under development. More information would be provided in the following version of the deliverable, D7.7.

2.5.8 Recommendations for improvement, future actions and process changes

In this section recommendations for the improvement of the tool are presented and the adopted future actions taken to enhance the tool, even within the eDREAM project lifetime. Regarding this tool no recommendation have been reported at this stage. This information would be updated in the next version of the deliverable.

2.6 Load Profiling

2.6.1 Functional Description

The Load Profiling module aims at categorizing users' profiles (consumption dataset of customer from the Italian pilot site) according to some pre-defined features. Basically, this module is able to organize the raw dataset (already pre-processed and filtered) according to some temporal features in order to achieve a better organization of the energy profiles and facilitate further analytics provided by other modules in the so-called Big Data Layer of the eDREAM platform. Nine features have been selected for this purpose: daily profiles, daily profiles in working days, daily profiles for weekends and holidays, weekly profiles on hourly basis, weekly profiles on daily basis and monthly profiles of the four seasons on daily basis.

2.6.2 Connection with other Components and Interfaces

The following table presents the necessary information about the implementation and integration stages, so as to achieve adequate connection and communication.

Table 28. Component Connection Requirements.

Component	***Connection Type***	***API Protocol***	***Data Type***	***Comments***
Input/output components connections are implemented within this module.	HTTP(GET/POST)	RESTful Services	JSON	-

2.6.3 Data Flow

The component receives inputs from the historical dataset of pilot site in the Decentralized Repository and the Electricity consumption/production forecasting module. This component provides output to the Big Data Clustering at multiple scale.

2.6.4 Integration Status

The information related to the integration process is mentioned in the following table in a high-level format without many details.

Table 29. Integration process Information.

Integration Status	Final version
Format for Integration	Web service application
Progress up to date	Communication with Decentralized Repository, Electricity Consumption/Production forecasting and Big Data Clustering at multiple scale.
Pending Integration Actions	Final version

2.6.5 Development Status

The below table depicts the relevant information with the development process.

Table 30. Development process information.

Development Status	Final
Programming Language	Python
Progress up to date	The main functionalities of the component have been implemented and tested within the data flow described in the Use Case3.1
Pending Development Actions	None

2.6.6 Software Problems Encountered

The characteristics of the software problems encountered are summarized in this section. With respect ot this tool, no software problems have been reported so far.

2.6.7 Functional performance and efficiency report

A brief description of the results of the verification process, including observations and conclusions from the feedback obtained by the involved actors, is presented in the following table.

Table 31. Performance test results.

Performance Requirement	Description	Remarks (Positive/Neutral/Negative)
MF02_BR02_UR09_NFR01 [1]	Be able to operate in a scalable way depending on the demands related to data injection, processing and storage	Positive
Retrieving Data	The system must be able to retrieve data from a great number of devices and services.	Positive
Modules Communication	the eDREAM platform modules must be able to communicate with each other to provide the expected functionalities	Positive
Modularity	The system must be composed of several micro-services, so as to better perform different processes on multiple machines.	Positive

2.6.8 Recommendations for improvement, future actions and process changes

In the following table recommendations for the improvement of the tool are presented and the adopted future actions taken to enhance the tool, even within the eDREAM project lifetime. This would be further clarified in the next version of the deliverable.

Table 32. Improvement Recommendations.

Recommendations for improvement	Future actions	Description
Customizable features	Improve the feature extractions with new options	We are adapting the module in order to allow customizable feature extraction, with particular regard of input from Electricity consumption/production forecasting

2.7 Big Data Clustering at Multiple Scales

2.7.1 Functional Description

The Big Data Clustering at Multiple Scale is aimed at grouping the data from users in well-defined and separated clusters, according to specific objectives. This component includes different algorithms that can be activated based on the amount of the input data, allowing to cluster the data with the most suitable technique. When a larger dataset is injected, a dedicated deep-learning algorithm is activated. This algorithm is used for better performance when big data and provides a more accurate clustering of the prosumers.

2.7.2 Connection with other Components and Interfaces

The following table presents the necessary information about the implementation and integration stages, so as to achieve adequate connection and communication.

Table 33. Component Connection Requirements.

Component	*Connection Type*	*API Protocol*	*Data Type*	*Comments*
Input/output components connections are implemented within this module.	HTTP(GET/POST)	RESTful Services	JSON	-

2.7.3 Data Flow

The component receives inputs from the historical dataset of pilot site in the Decentralized Repository and the Load Profiles module, both stored in a Cassandra DB. This component is also connected to the Customer Segmentation module, providing the results of clusterization and receiving the segmentation of the customers. Finally, the Big Data Clustering at Multiple Scale module will receive requests from the Graph-Based Analytics and the Decision Support System and Demand Response Strategies Optimization UI asking for the clusters of prosumers.

2.7.4 Integration Status

The information related to the integration process is mentioned in the following table in a high-level format without many details.

Table 34. Integration process Information.

Integration Status	Final version
Format for Integration	Web service application
Progress up to date	Communication with Decentralized Repository, Load Profiling and Customer Segmentation.
Pending Integration Actions	Final version

2.7.5 Development Status

The below table depicts the relevant information with the development process.

Table 35. Development process information.

Development Status	Final
Programming Language	Python
Progress up to date	The main functionalities of the component have been implemented and tested within the data flow described in the Use Cases 1.6 and 3.1 [1]
Pending Development Actions	None

2.7.6 Software Problems Encountered

The characteristics of the software problems encountered are summarized in this section. With respect ot this tool, no software problems have been reported so far.

2.7.7 Functional performance and efficiency report

A brief description of the results of the verification process, including observations and conclusions from the feedback obtained by the involved actors, is presented in the following table.

Table 36. Performance test results.

Performance Requirement	**Description**	**Remarks (Positive/Neutral/Negative)**
MF02_BR01_UR01_NFR01 [1]	Be able to operate in a scalable way depending on the demands related to data injection, processing and storage	**Positive**

MF02_BR01_UR01_NFR02 [1]	Be able to decompose the solution in different microservices, so as to run better during different processes in several machines	**Positive**
MF02_BR01_UR03_NFR03 [1]	Be able to document information types produced, consumed and transformed in an information model which shall include the relationships between information types	**Neutral**
MF02_BR01_UR04_NFR04 [1]	Be able to process data gathered from different sources in order to achieve flexibility profiling. It is crucial for such calculation to ensure the capacity to provide data coming from differed database and data lake (batch, preprocessed, other modules outputs, devices etc.).	**Positive**
MF02_BR01_UR05_NFR05 [1]	Information models that govern the data exchanged with the different types of devices and managed or stored by the modules will consider context data or metadata (e.g. location, accuracy, ownership etc.)	**Positive**
Ease of use case #1.6 and use case#3.1 [1]	The messages provided by the tool must be clear and easy to understand	The load profiling does not return any messages to the user rather than the output graph **Neutral**

2.7.8 Recommendations for improvement, future actions and process changes

In this section recommendations for the improvement of the tool are presented and the adopted future actions taken to enhance the tool, even within the eDREAM project lifetime. Regarding this tool no recommendation have been reported at this stage. This information would be updated in the next version of the deliverable.

2.8 Customer Segmentation

2.8.1 Functional Description

The Customer Segmentation module is aimed at recognizing a consumer's load profile pattern that had been clustered in Big Data Clustering at multiple scale module. In that way, when new consumers are added to the aggregator's portfolio, via this tool they are assigned directly to the most suitable cluster.

2.8.2 Connection with other Components and Interfaces

The following table presents the necessary information of the implementation and integration stages, so as to achieve adequate connection and communication.

Table 37. Component Connection Requirements.

Component	***Connection Type***	***API Protocol***	***Data Type***	***Comments***
Input/output components connections are implemented within this module.	HTTP(GET/POST)	RESTful Services	JSON	-

2.8.3 Data Flow

The Customer Segmentation module is connected with the Big Data Clustering at multiple scale module. It receives the calculated cluster from the Big Data Clustering module and sends back the information of the prosumers' profiles with the assigned cluster.

2.8.4 Integration Status

The information related to the integration process is mentioned in the following table in a high-level format without many details.

Table 38. Integration process Information.

Integration Status	Final version
Format for Integration	Web service application
Progress up to date	Communication with Decentralized Repository, Load Profiling and Customer Segmentation.
Pending Integration Actions	Final version

2.8.5 Development Status

The below table depicts the relevant information with the development process.

Table 39. Development process information.

Development Status	Final
Programming Language	Python
Progress up to date	The main standalone functionalities of the component have been implemented
Pending Development Actions	Unit testing

2.8.6 Software Problems Encountered

The characteristics of the software problems encountered are summarized in this section. With respect ot this tool, no software problems have been reported so far.

2.8.7 Functional performance and efficiency report

A brief description of the results of the verification process, including observations and conclusions from the feedback obtained by the involved actors, is presented in the following table.

Table 40. Performance test results..

Performance Requirement	**Description**	**Remarks (Positive/Neutral/Negative)**
MF02_BR02_UR09_NFR01 [1]	Be able to operate in a scalable way depending on the demands related to data injection, processing and storage	**Positive**
Ease of use case#3.1 [1]	The messages provided by the tool must be clear and easy to understand	The load profiling does not return any messages to the user rather than the output graph **Neutral**
Modules Communication	the eDREAM platform modules must be able to communicate with each other to provide the expected functionalities	**Positive**
Modularity	The system must be composed of several micro-services, so as to better perform different processes on multiple machines.	**Positive**

2.8.8 Recommendations for improvement, future actions and process changes

In this section recommendations for the improvement of the tool are presented and the adopted future actions taken to enhance the tool, even within the eDREAM project lifetime. Regarding this tool no recommendation have been reported at this stage. This information would be updated in the next version of the deliverable.

2.9 VPP & DR Services Optimization engine

2.9.1 Functional Description

This tool is part of the optimization process, and in particular aids in formulating the optimal DR scheduling of the aggregator's portfolio registered assets, that is Distributed Energy Resources (DERs), or Microgrids etc. It utilizes mainly a tailored economic dispatch and unit commitment problem process. More information is presented in Deliverable D4.1 [6].

2.9.2 Connection with other Components and Interfaces

The following table presents the necessary information of the implementation and integration stages, so as to achieve adequate connection and communication.

Table 41. Component Connection Requirements.

Component	***Connection Type***	***API Protocol***	***Data Type***	***Comments***
Input components connections: - Electricity Consumption/Production Forecasting - PV/RES Degredation & Trend Analysis - Baseline Flexibility Esitmation - Virtual Power Plants, Generation, Modeling & Forecasting - Cassandra DB Output Components connections: - DSS & DR Strategies Optimization	TCP/IP, HTTP	RESTful services	JSON	-

2.9.3 Data Flow

This tools retrieves data of various other tools. More specifically, forecasted data from the Electricity consumption/production Forecasting and the PV/RES Degradation & Trend Analysis tools; The Prosumers' baseline load flexibility from the Baseline Flexibility Estimation tool; Optimal Coalitions from Virtual Power Plants Generation, Modeling & Forecasting tool; and, finally, KPIs and necessary constraints from Cassandra DB. The output results, that is information regarding the optimal DR scheduling of consumers, prosumers, DERs and Microgrids are fed to the DSS & DR Strategies Optimization tool.

2.9.4 Integration Status

The information related to integration process are mentioned in the following table in a high-level format without many details.

Table 42. Integration process Information.

Integration Status	Under Development
Format for Integration	Web service application
Progress up to date	Communication with Cassandra DB and DSS & Strategies Optimization has been developed
Pending Integration Actions	Communication with the rest of the related tools Interfaces to be tested in real case by testing the deployment version on the pilots.

2.9.5 Development Status

The below table depicts the relevant information with the development process.

Table 43. Development process information.

Development Status	Under Development
Programming Language	Python
Progress up to date	The main standalone functionalities of the component have been implemented
Pending Development Actions	The functionalities related to other components have to be finalized

2.9.6 Software Problems Encountered

The characteristics of the software problems encountered are summarized in this section. With respect ot this tool, no software problems have been reported so far.

2.9.7 Functional performance and efficiency report

A brief description of the results of the verification process, including observations and conclusions from the feedback obtained by the involved actors, is presented in this section. In the case of this tool no performance tests have been conducted yet, since it is under development. More information would be provided in the following version of the deliverable, D7.7.

2.9.8 Recommendations for improvement, future actions and process changes

In this section recommendations for the improvement of the tool are presented and the adopted future actions taken to enhance the tool, even within the eDREAM project lifetime. Regarding this tool no recommendation have been reported at this stage. This information would be updated in the next version of the deliverable.

2.10 Distributed Ledger

2.10.1 Functional Description

The storage solution implements a storage mechanism to handle data from the smart meters over BigchainDB. Smart Meters have been modeled as *Assets* and the energy data gathered from the smart meters are included as transaction metadata. More details are provided in the eDREAM deliverable D5.1 [7].

2.10.2 Connection with other Components and Interfaces

The following table presents the necessary information about the implementation and integration stages, so as to achieve adequate connection and communication.

Table 44. Component Connection Requirements.

Component	***Connection Type***	***API Protocol***	***Data Type***	***Comments***
Field devices	Publisher/Subscribe	HTTP POST	Smart meter data	The */notify* endpoint is used to store smart meter data using BigchainDB. A subscription on Orion Context Broker has to be created so, once the conditions of the subscription have been met, a POST request containing all affected entities will be sent to the endpoint, which handles the notification and provides to the creation of the right CREATE or TRANSFER transaction

2.10.3 Data Flow

The data flow of this tool is depicted in the following figure.

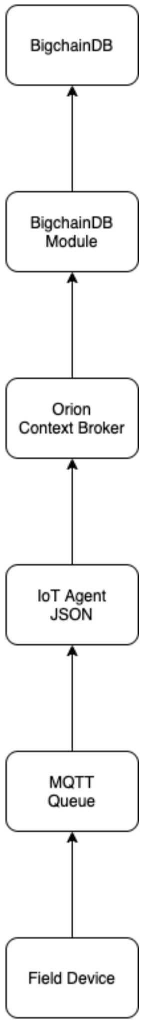

Figure 5. Data flow diagram of the Distributed Ledger tool.

2.10.4 Integration Status

The information related to the integration process is mentioned in the following table in a high-level format without many details.

Table 45. Integration process Information.

Integration Status	Fully integrated in CI/CD environment
Format for Integration	YAML configuration file for Rancher orchestration platform
Progress up to date	Configured for continuous integration and automatic deployment
Pending Integration Actions	-

2.10.5 Development Status

The following table depicts the relevant information with the development process.

Table 46. Development process information.

Development Status	Released
Programming Language	NodeJS
Progress up to date	1st version released in May 2019
Pending Development Actions	2nd version expected to be released in May 2020

2.10.6 Software Problems Encountered

The characteristics of the software problems encountered are summarized in this section. With respect ot this tool, no software problems have been reported so far.

2.10.7 Functional performance and efficiency report

A brief description of the results of the verification process, including observations and conclusions from the feedback obtained by the involved actors, is presented in the following table.

Table 47. Performance test results.

Performance Requirement	Description	Remarks (Positive/Neutral/Negative)
MF03_BR01_UR02_NFR01 [1]	Ensure scalability with high number of energy transactions	**Positive**
MF03_BR01_UR03_NFR02 [1]	Store transactions in a secured and tamperproof manner	**Positive**
MF03_BR01_UR04_NFR03 [1]	Grant access to data only to authorized users	**Positive**

2.10.8 Recommendations for improvement, future actions and process changes

In the following table recommendations for the improvement of the tool are presented and the adopted future actions taken to enhance the tool, even within the eDREAM project lifetime. This would be further clarified in the next version of the deliverable.

Table 48. Improvement Recommendations.

Recommendations for improvement	Future actions	Description
Extend supported data formats	To be integrated in 2nd release	Support the different data formats used from pilot's smart meters / EVs / charging stations

2.11 Blockchain-driven control for LV networks

2.11.1 Functional Description

Each flexibility asset (Prosumer/ DSO/ Aggregator) in the grid is enabled to register via smart contracts his flexibility offer/ requests. All the orders are gathered during the market session period, and by the end of the market session the flexibility orders are matched, and the matched prosumers are requested to deliver the promised flexibility profiles. During real-time monitoring, the activity of each prosumer is evaluated and financially settled.

2.11.2 Connection with other Components and Interfaces

The following table presents the necessary information about the implementation and integration stages, so as to achieve adequate connection and communication.

Table 49. Component Connection Requirements.

Component	***Connection Type***	***API Protocol***	***Data Type***	***Comments***
Input connections from *Electricity Production/Consumption Forecasting* and *Baseline Flexibility Estimation* Components	TCP/IP, HTTP, etc.	RESTful Services	JSON	Flexibility forecasts and Baseline estimation profiles are required in order to have the flexibility potential of the prosumer, as reference profiles for their real-time activity.
Input connections from *Secured Data Handling Through Ledger* component	JSON-RPC	Web3	JSON	The values monitored by sensors are forwarded to the Distributed Ledger, and are relevant for the evaluation of the prosumer's activity in real-time.
Output connections to the *Closed Loop DR Verification Engine* component	-	-	Custom Internal Structures used by the Smart Contracts	The real-time evaluation of the Prosumer activity is forwarded to the Closed Loop DR Verification Engine, for validation and financial settlement.

2.11.3 Data Flow

Upon registration, the prosumer needs to set its flexibility potential (the estimated prosumer capacity to lower the energy consumption, the estimated prosumer capacity to increase the energy consumption, and the baseline as the regular energy consumption of the prosumer). The prosumer is evaluated in near-real-time for its flexibility provided, based on the flexibility potential and the flexibility requests set during the day-ahead markets, by computing the deviation between the registered monitored values and the baseline consumption.

2.11.4 Integration Status

The information related to the integration process is mentioned in the following table in a high-level format without many details.

Table 50. Integration process Information.

Integration Status	Under development
Format for Integration	Web Service Application; Smart Contracts
Progress up to date	Smart Contracts have been developed and integrated. RESTful APIs are exposed as a middleware between the external components and the Smart Contracts APIs
Pending Integration Actions	Interfaces to be tested in real case

2.11.5 Development Status

The below table depicts the relevant information with the development process.

Table 51. Development process information.

Development Status	Final
Programming Language	Solidity; NodeJS
Progress up to date	The main standalone functionalities of the component have been implemented
Pending Development Actions	None

2.11.6 Software Problems Encountered

The characteristics of the software problems encountered are summarized in this section. With respect ot this tool, no software problems have been reported so far.

2.11.7 Functional performance and efficiency report

A brief description of the results of the verification process, including observations and conclusions from the feedback obtained by the involved actors, is presented in this section. In the case of this tool no performance tests have been conducted yet, since it is under development. More information would be provided in the following version of the deliverable, D7.7.

2.11.8 Recommendations for improvement, future actions and process changes

In this section recommendations for the improvement of the tool are presented and the adopted future actions taken to enhance the tool, even within the eDREAM project lifetime. Regarding this tool no recommendation have been reported at this stage. This information would be updated in the next version of the deliverable.

2.12 Secured Blockchain-driven Energy Market

2.12.1 Functional Description

The peer to peer energy marketplace is implemented at the local micro-grid level and enacts any prosumer regardless of size to directly participate in the market trading sessions. The prosumers will use the energy forecasts to estimate their production/ consumption and participate in the electricity market sessions. They will leverage on self-enforcing smart contracts to submit the bids/offers and transact in a peer to peer fashion.

2.12.2 Connection with other Components and Interfaces

The following table presents the necessary information about the implementation and integration stages, so as to achieve adequate connection and communication.

Table 52. Component Connection Requirements.

Component	***Connection Type***	***API Protocol***	***Data Type***	***Comments***
Input connections from *Electricity Production/Consumption Forecasting* and *Baseline Flexibility Estimation* Components	TCP/IP, HTTP, etc.	RESTful Services	JSON	Energy forecasts are required in order to have an estimation of the energy that the prosumer can sell/buy from the market in the following period.
Input connections from *Secured Data Handling Through Ledger* component	JSON-RPC	Web3	JSON	The values monitored by sensors are forwarded to the Distributed Ledger, and are relevant for the evaluation of the prosumer's activity in real-time.
Input/output connections to the *Local/Remote HMI*	TCP/IP, HTTP, etc.	RESTful Services	JSON	The prosumer is allowed to evaluate and customize the order details regarding the Energy Bids/ Offers registered in the Energy Market.

Output connections to the *Closed Loop DR Verification Engine*	-	-	Custom Internal Structures used by the Smart Contracts	The real-time evaluation of the Prosumer activity is forwarded to the Closed Loop DR Verification Engine, for validation and financial settlement.

2.12.3 Data Flow

Based on the forecasted energy production/consumption, a prosumer can register energy offers/bids for the following day in the P2P energy Market. The forecasts and the prosumer activity are depicted in the HMI. Based on its values registered in near-real-time the Prosumer's activity is evaluated against the matched offers/bids and if deviations are registered, the Prosumer is penalized.

2.12.4 Integration Status

The information related to the integration process is mentioned in the following table in a high-level format without many details.

Table 53. Integration process Information.

Integration Status	Under development
Format for Integration	Web Service Application; Smart Contracts
Progress up to date	Smart Contracts have been developed and integrated. RESTful APIs are exposed as a middleware between the external components and the Smart Contracts APIs
Pending Integration Actions	Interfaces to be tested in real case

2.12.5 Development Status

The below table depicts the relevant information with the development process.

Table 54. Development process information.

Development Status	Final
Programming Language	Solidity; NodeJS
Progress up to date	The main standalone functionalities of the component have been implemented
Pending Development Actions	None

2.12.6 Software Problems Encountered

The characteristics of the software problems encountered are summarized in this section. With respect ot this tool, no software problems have been reported so far.

2.12.7 Functional performance and efficiency report

A brief description of the results of the verification process, including observations and conclusions from the feedback obtained by the involved actors, is presented in the following table.

Table 55. Performance test results.

Performance Requirement	Description	Remarks (Positive/Neutral/Negative)
MF03_BR04_UR01_NFR01 [1]	Store transactions in a secured and tamperproof manner	Authorization and Authentication mechanisms are implemented for each operation exposed by the Smart Contracts. The orders (energy bids/offer) can be registered only by the prosumer in the P2P energy Market, by signing the order-publishing transactions. Upon registration, the sender's signature is validated against the authorized accounts (specific for each market operation). The tamperproof requirement is satisfied for each transaction mined in the chain, thus once the orders are published in the Smart Contracts, and the corresponding transaction is mined, the orders are immutable. **Positive**

2.12.8 Recommendations for improvement, future actions and process changes

In this section recommendations for the improvement of the tool are presented and the adopted future actions taken to enhance the tool, even within the eDREAM project lifetime. Regarding this tool no recommendation have been reported at this stage. This information would be updated in the next version of the deliverable.

2.13 Closed loop DR Verification engine

2.13.1 Functional Description

This component returns the coefficient to apply to the *basic* incentives/penalties for a prosumer in a given discrete time range, say [T0, T1, T2, ..., TN], to obtain the actual incentive/penalty. Coefficients can vary from a maximum of 1 (when the production request is exactly satisfied, meaning that the whole basic incentive is awarded) to potentially -∞. More details are provided in deliverable D5.3 [8] to be released in M28, i.e. April 2020.

2.13.2 Connection with other Components and Interfaces

The following table presents the necessary information about the implementation and integration stages, so as to achieve adequate connection and communication.

Table 56. Component Connection Requirements.

Component	***Connection Type***	***API Protocol***	***Data Type***	***Comments***
Blockchain-driven control for LV networks	Async	HTTP	Input: actual smart meter readings, requested profile, forecast (arrays of floats, one for each timestamp). Tolerance range R, Multiplier M.	Return an array of N coefficients that are related to [T_0, T_1, T_2, ..., T_N]

2.13.3 Data Flow

The data flow of this tool is depicted in the following figure.

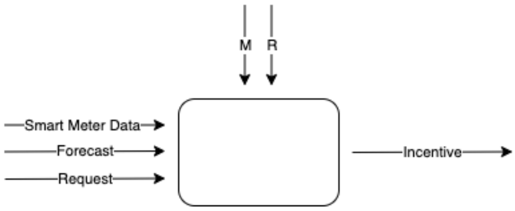

Figure 6. Data flow diagram for the Closed loop DR verification engine tool.

2.13.4 Integration Status

The information related to the integration process is mentioned in the following table in a high-level format without many details.

Table 57. Integration process Information.

Integration Status	Fully integrated in CI/CD environment
Format for Integration	YAML configuration file for Rancher orchestration platform
Progress up to date	Configured for continuous integration and automatic deployment
Pending Integration Actions	-

2.13.5 Development Status

The below table depicts the relevant information with the development process.

Table 58. Development process information.

Development Status	Released
Programming Language	Python
Progress up to date	Released in April 2020
Pending Development Actions	-

2.13.6 Software Problems Encountered

The characteristics of the software problems encountered are summarized in this section. With respect ot this tool, no software problems have been reported so far.

2.13.7 Functional performance and efficiency report

A brief description of the results of the verification process, including observations and conclusions from the feedback obtained by the involved actors, is presented in this section. In the case of this tool no performance tests have been conducted yet, since it is under development. More information would be provided in the following version of the deliverable, D7.7.

2.13.8 Recommendations for improvement, future actions and process changes

In this section recommendations for the improvement of the tool are presented and the adopted future actions taken to enhance the tool, even within the eDREAM project lifetime. Regarding this tool no recommendation have been reported at this stage. This information would be updated in the next version of the deliverable.

2.14 Graph-based Analytics - HMIs

2.14.1 Functional Description

The Graph-based Analytics component delivers a multi-purpose visualization tool utilizing various graphs. The component received data regarding prosumers from the pilot sites and transforms them in order to produce the required graphs. Since there are very close ties and interconnection between the Graph-based Analytics and the Human Machine Interface (HMIs), these two tool sets, are presented together. The main views of these components can be summarized as follows:

- The *geolocation of the prosumers* (data is artificially generated within Terni area which is one of our pilot sites)
- *KPIs* view which showcases the comparison of multiple user KPIs such as CO2_reduction or Demand Flexibility
- *Trend Analysis* output for a specific timeslot
- *Multi-Objective analysis V1* for a specific timeslot showing the individual and merged graphs based on specific prosumer feature selection.
- *Multi-Objective analysis V2* that allows feature selection to plot prosumer with similar feature values.
- *K-partite analysis* that bins the users according to specific features based on a datetime filter.
- *Geographical Analysis (Historic)* that produces a heatmap of consumptions according to user datetime request.
- *Geographical Analysis (Day-Ahead)* that forms clusters of close vicinity aggregated consumptions of multiple prosumers based on historic data.
- *Optimal Scheduling Clustering Analysis.*

These views offer various functionalities such as:

- Provide data visualization in different granularities (scale/time), on large volumes of historical data as well as processed data.
- Provide efficient connectivity and communication through RESTful interfaces (mainly with the prosumers database).

2.14.2 Connection with other Components and Interfaces

The following table presents the necessary information about the implementation and integration stages, so as to achieve adequate connection and communication with other components.

Table 59. Component Connection Requirements.

Component	*Connection Type*	*API Protocol*	*Data Type*	*Comments*
Input/output components connections	TCP/IP, HTTP	RESTful Services	JSON, CSV	Receive data from the Cassandra DB.

2.14.3 Data Flow

The Graph based analytics module receives data (for the time being, historical) regarding prosumers' consumption/production from the Cassandra DB and feeds them to the front-end in order to produce the multi-purpose dashboard.

2.14.4 Integration Status

The information related to integration process are mentioned in the following table in a high-level format without many details.

Table 60. Integration process Information.

Integration Status	Under development
Format for Integration	Windows application/Web service application
Progress up to date	Communication with Cassandra DB and services that produce the graphs
Pending Integration Actions	The whole component to be tested in real cases deploying the multi-purpose dashboard to the pilots.

2.14.5 Development Status

The below table depicts the relevant information with the development process.

Table 61. Development process information.

Development Status	Under development
Programming Language/Frameworks	Python-JavaScript/Angular 8/NodeJS
Progress up to date	The main standalone functionalities of the component have been implemented
Pending Development Actions	The functionalities related to other components have to be finalized as well as the multi-purpose dashboard view to be finalized

2.14.6 Software Problems Encountered

The characteristics of the software problems encountered are summarized in the following table.

Table 62. Software Problems characteristics

Failure Type	Failure Description	Failure Cause	Countermeasure
Slow Response Time when passing the data to form the graphs	The system takes a timeout error	Slow network, or large volume of data	Parse data incrementally or aggregate the data before visualizing

2.14.7 Functional performance and efficiency report

A brief description of the results of the verification process, including observations and conclusions from the feedback obtained by the involved actors, is presented in the following table.

Table 63. Performance test results.

Performance Requirement	Description	Remarks (Positive/Neutral/Negative)
Average response time	The time that is required to get the visual output since the user request	The response time varies between 2-5 seconds depending on the request time period to be fed to the graph generation algorithms. For input time periods from 1 day to 5 days the average response time is 3 seconds. Neutral
Error rates	**HTTP Error %:** Number of web requests that ended in an error. We performed 100 requests varying from 1 to 3 days of historic data from the nZEB platform data resulting in 2 HTTP error throws. **Result** 2% HTTP error. **Logged Exceptions:** Number of unhandled and logged errors from your application. After running 100 requests there were no unhandled and logged errors. **Thrown Exceptions:** Number of all exceptions that have	Positive

	been thrown. After running 100 requests there were no exceptions thrown.	
System Availability	The system ensures availability of services 99%, or superior at any time, 24h/24h, 7d/7d, independently from performances of other applications.	Positive
MF02_BR05_BR10_MF03_BR05_NFR01 [1]	The User Interface shall have a user-friendly look, fully customized to the needs of different stakeholders	Positive
MF02_BR05_BR10_MF03_BR05_NFR02 [1]	The User Interface shall provide a user interface offering maps Visualization for a more concrete analysis	Positive
MF02_BR05_BR10_MF03_BR05_NFR03 [1]	The User Interface shall be able to allow an easy discoverability of the actions available	Positive
MF02_BR05_BR10_MF03_BR05_NFR04 [1]	The User Interface shall be tailored to the end user needs	Positive
MF02_BR05_BR10_MF03_BR05_NFR05 [1]	The messages provided by the system must be clear and easy to understand	The graph analytics view does not return any messages to the user rather than the output graph visualization Neutral
MF02_BR05_BR10_MF03_BR05_NFR06 [1]	The User Interface must be simple and intuitive	Positive

2.14.8 Recommendations for improvement, future actions and process changes

In the following table recommendations for the improvement of the tool are presented and the adopted future actions taken to enhance the tool, even within the eDREAM project lifetime. This would be further clarified in the next version of the deliverable.

Table 64. Improvement Recommendations.

Recommendations for improvement	Future actions	Description
Improve response times regarding the output of algorithm that forms the graphs	Update the data storing process so that the front-end can request the data faster	We can set a weekly historic data storage frequency so that the database of the graph-based analytics component can have prestored the requested data

2.15 DSS & DR Strategies Optimization

2.15.1 Functional Description

This tool is the main decision-making tool of the eDREAM platform. All the necessary input from the various tools before being fed to the visualization platform, are passed through this tool, along with the appropriate input from the end user itself, being one of the stakeholders of the defined eDREAM ecosystem. This tool analyzes and finds the feasible optimal solutions, in order for a major decision to be determined by one of the involved stakeholders. Once they are reached, they are depicted via the visualization platform and the end user decides the course of action to be taken.

2.15.2 Connection with other Components and Interfaces

The following table presents the necessary information about the implementation and integration stages, so as to achieve adequate connection and communication.

Table 65. Component Connection Requirements.

Component	*Connection Type*	*API Protocol*	*Data Type*	*Comments*
Input from: PV/RES Degradation & Trend Analysis, Graph-based analytics, Big data Clustering at multiple scales, Cassandra DB Output to: Graph-based Analytics	HTTP(GET/POST)	RESTful Services	JSON	Receives data from all appropriate tools, in order to provide the feasible optimal solutions for each problem set by each of the involved stakeholders within the eDREAM ecosystem.

2.15.3 Data Flow

Data concerning end-user preferences, forecasted electrical consumption energy prices, prosumers clustered results are fed into the tool and the feasible optimal solutions are presented to the final user via the visualization platform.

2.15.4 Integration Status

The information related to the integration process is mentioned in the following table in a high-level format without many details.

Table 66. Integration process Information.

Integration Status	Under development
Format for Integration	Windows application/Web service application
Progress up to date	Communication with the other tools and services
Pending Integration Actions	The whole component to be tested in real cases

2.15.5 Development Status

The below table depicts the relevant information with the development process.

Table 67. Development process information.

Development Status	Under development
Programming Language	Python
Progress up to date	The main standalone functionalities of the component have been implemented
Pending Development Actions	The functionalities related to other components have to be finalized

2.15.6 Software Problems Encountered

The characteristics of the software problems encountered are summarized in this section. With respect ot this tool, no software problems have been reported so far.

2.15.7 Functional performance and efficiency report

A brief description of the results of the verification process, including observations and conclusions from the feedback obtained by the involved actors, is presented in this section. In the case of this tool no performance tests have been conducted yet, since it is under development. More information would be provided in the following version of the deliverable, D7.7.

2.15.8 Recommendations for improvement, future actions and process changes

In this section recommendations for the improvement of the tool are presented and the adopted future actions taken to enhance the tool, even within the eDREAM project lifetime. Regarding this tool no recommendation have been reported at this stage. This information would be updated in the next version of the deliverable.

2.16 DR Aerial Survey Toolkit

2.16.1 Functional Description

In this task, an image processing algorithm that produced estimates for various features related to Demand Response estimation is developed. The image inputs are expected to be a series of frames in the optical and infrared bands, captured by an Unmanned Aerial Vehicle externally surveying a building of interest. Estimated features include information on the position of the building, thermography statistics, information on windows & doors of the structure, photovoltaic elements and other structural assets.

2.16.2 Connection with other Components and Interfaces

The following table presents the necessary information about the implementation and integration stages, so as to achieve adequate connection and communication.

Table 68. Component Connection Requirements.

Component	*Connection Type*	*API Protocol*	*Data Type*	*Comments*
Visualization front-end	TCP/IP	RESTful API	JSON	The image processing tool produces output that can be either printed on the working terminal as text information, or sent via a REST / POST command to the server running the visualization front-end as a JSON.

2.16.3 Data Flow

Optical and infrared inputs are used to produce deep features, using a deep neural network architecture. Deep features are used to estimate the structure-of-interest position at a pixel-level. This enables using the position to estimate a mask on the infrared input, thereby producing accurate thermography statistics. A second neural network is used to further process deep features and raw optical data in order to detect and calculate positional estimates for the stated structural assets. The output of this latter network is further post-processed to produce Boolean indicators (existence/non-existence) of assets as well as numerical estimates (number/area of assets).

2.16.4 Integration Status

The information related to integration process are mentioned in the following table in a high-level format without many details.

Table 69. Integration process Information.

Integration Status	Under development
Format for Integration	RESTful API client
Progress up to date	Finalization of the required JSON packet components and overall communication protocol.
Pending Integration Actions	Test sending the feature estimates via JSON and REST commands on pilot site use-cases.

2.16.5 Development Status

The below table depicts the relevant information with the development process.

Table 70. Development process information.

Development Status	Under development
Programming Language	Python / Numpy / Scipy / Tensorflow
Progress up to date	An image processing tool that comprises the following sub-components: Semantic annotation, Structure-of-interest position estimation, Asset feature extraction, Feature post-processing.
Pending Development Actions	Completion of annotation, further training over deep feature inputs, further post-processing on deep features and neural network outputs. Running tests in the context of quality control.

2.16.6 Software Problems Encountered

The characteristics of the software problems encountered are summarized in the following table.

Table 71. Software Problems characteristics.

Failure Type	Failure Description	Failure Cause	Countermeasure
Originally planned system component replaced with less computationally expensive components	Structure-from-motion based estimates dropped in favor of 2D image processing	Computational cost of Structure-from-Motion	Use of 2D image processing techniques

2.16.7 Functional performance and efficiency report

A brief description of the results of the verification process, including observations and conclusions from the feedback obtained by the involved actors, is presented in the following table.

Table 72. Performance test results.

Performance Requirement	Description	Remarks (Positive/Neutral/Negative)
Estimation of thermal statistics, thermal index	Evaluation of proposed method for estimation of thermal signature of the structure of interest	Positive
Detection of photovoltaic elements on structure roof	Evaluation of proposed method for detection of photovoltaic elements on structure roof	Positive

2.16.8 Recommendations for improvement, future actions and process changes

In the following table recommendations for the improvement of the tool are presented and the adopted future actions taken to enhance the tool, even within the eDREAM project lifetime. This would be further clarified in the next version of the deliverable.

Table 73. Improvement Recommendations.

Recommendations for improvement	Future actions	Description
Evaluate additional metrics, with numerical experiments over feature extraction estimates	Run evaluation metrics	Numerical evaluation of detection efficiency of DR-related asset features

2.17 Forecasting Tool

2.17.1 Functional Description

The main functionality of the Forecasting Tool is to provide an interactive visualization framework enabling the aggregator to visualize the electricity consumption/production/flexibility historical and forecasted data as well as the baseline for each energy prosumer over a given day at a selected timeframe (day-ahead, intraday, near-real-time) and granularity (one hour or 30 minutes).
The main functionalities of these components can be summarized as follows:

- Allow the selection of a prosumer, a timeframe and a granularity;
- Provide baseline estimation for the selected prosumer;
- Provide energy consumption/production/flexibility forecasts for the selected prosumer, timeframe and granularity;

2.17.2 Connection with other Components and Interfaces

The following table presents the necessary information about the implementation and integration stages, to achieve adequate connection and communication.

Table 74. Component Connection Requirements.

Component	*Connection Type*	*API Protocol*	*Data Type*	*Comments*
Input components connections: - *Cassandra DB*	TCP/IP, HTTP, etc.	RESTful Services, MQTT, etc.	JSON	The output of the component is displayed on an *End-User Web Interface*

2.17.3 Data Flow

The input of the components is taken from the Cassandra DB where it was saved previously by the Baseline Flexibility Estimation and Electricity consumption/Production Forecasting components. Prosumer related data, such as historical monitored data, baseline load estimations (Baseline load flexibility and Baseline load fluctuations for enabling DR program application) and energy consumption/production/flexibility forecasted data at day-ahead, intra-day and near-Realtime timeframes are loaded from the database for a given prosumer ID and a selected date, are rendered and displayed as charts on the web interface.

2.17.4 Integration Status

The information related to the integration process is mentioned in the following table in a high-level format without many details.

Table 75. Integration process Information.

Integration Status	Final
Format for Integration	Web service application deployed in Docker container
Progress up to date	Communication with database to retrieve all the necessary data was implemented
Pending Integration Actions	None

2.17.5 Development Status

The below table depicts the relevant information with the development process.

Table 76. Development process information.

Development Status	Final
Programming Language	ReactJS
Progress up to date	All the functionality of the application has been developed
Pending Development Actions	None

2.17.6 Software Problems Encountered

The characteristics of the software problems encountered are summarized in this section. With respect ot this tool, no software problems have been reported so far.

2.17.7 Functional performance and efficiency report

A brief description of the results of the verification process, including observations and conclusions from the feedback obtained by the involved actors, is presented in this section. In the case of this tool no performance tests have been conducted yet, since it is under development. More information would be provided in the following version of the deliverable, D7.7.

2.17.8 Recommendations for improvement, future actions and process changes

In this section recommendations for the improvement of the tool are presented and the adopted future actions taken to enhance the tool, even within the eDREAM project lifetime. Regarding this tool no recommendation have been reported at this stage. This information would be updated in the next version of the deliverable.

3 Hardware deployment

In this section the deployment view is presented, as depicted first in the eDREAM Deliverable D2.5 [2]. The Deployment View presents aspects of the system that are connected with the realization of the system's components in the physical world. This view defines the physical entities of the environment, in which the system is intended to perform its running processes and operations, including:

- Technical environment (e.g. processing nodes, network interconnections, etc.);
- Mapping of software elements to the runtime environment;
- Third-party software requirements;
- Network requirements.

This architectural view will provide a first overview of the deployment environment of the eDREAM platform, which depends on the topology of the two pilot sites, covering the currently known hardware requirements of the software modules and the tools to be used represented in UML Deployment Diagram.

The two pilot areas comprise a variety of energy assets and smart metering devices that will provide the necessary real-time measurements for the testing of the eDREAM platform under different operating scenarios and conditions. The main infrastructure and the current operations of the two pilot sites, that will constitute the Field Data Aggregation layer of the platform and will determine the communication protocols with the field devices, are described in brief in the following two sections.

3.1 Active Micro-Grid

The main infrastructure of Terni's pilot site is an urban microgrid equipped with devices enabling the application of DR programs. This microgrid is connected to a secondary substation of the ASM electric grid, including four blocks of energy units as depicted in the figure below:

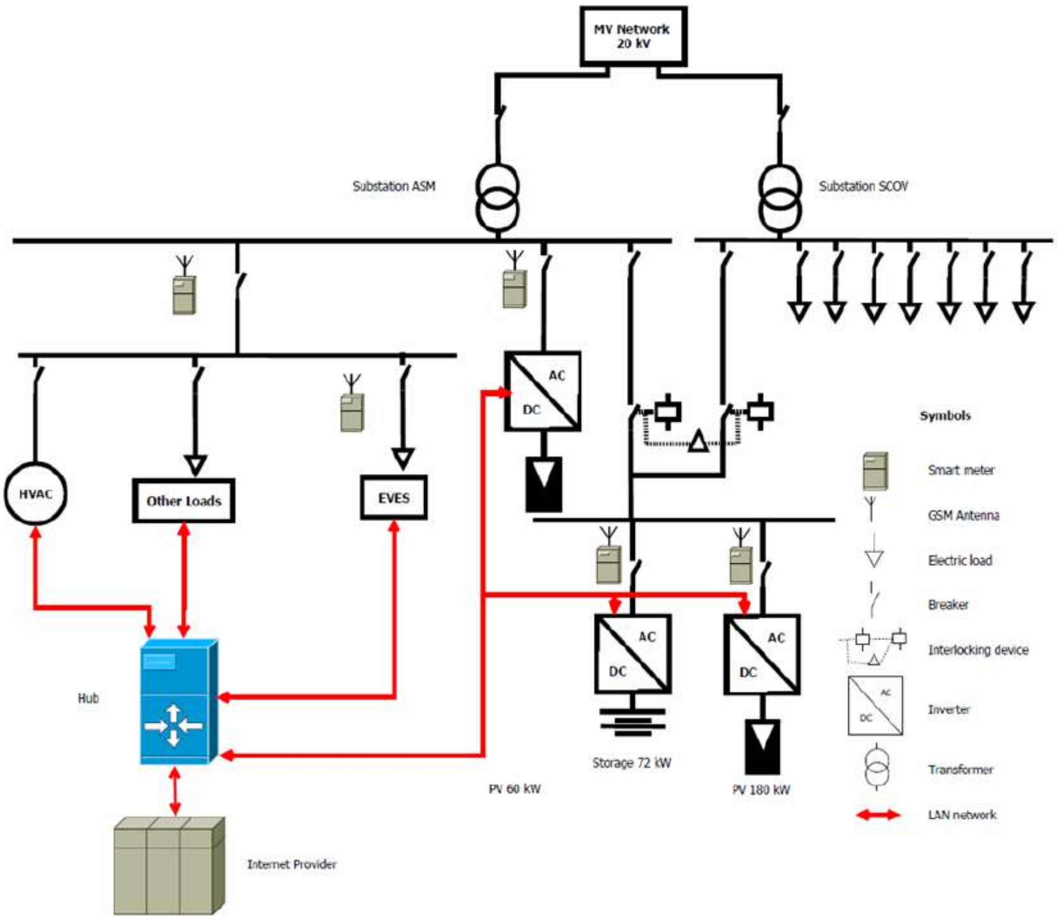

Figure 7. Internal and external connections in the Terni Micro-Grid

The ASM electric grid is characterized by a large number of distributed renewable energy sources embedded in the medium and low voltage distribution networks. It is worth pointing out that the measurements provided by the ASM electric grid for the year 2016 indicate that the yearlyReverse Power flow (RPF) measured in the substations reached 25 GWh and about 40% of the measurements show that the locally-produced energy overcomes energy drawn from the NTG (National Transmission Grid). Thus, towards avoiding the bidirectional power flow, the priority is to balance the Distributed Generation (DG) with the local consumption. One way to achieve this is the creation of secure and resilient microgrid environments through the participation in DR programs.

In order to get real time measurements and support real time operations in a Smart Grid environment, the Terni Micro-Grid is equipped with advanced smart meter technologies, such as the following:

- 3-phase ZMD meters (Landys+Gyr);
- Class A power quality analyzer WALLY A-RTU (3-phase high-precision analyzer and recorder, power quality, power meter, fault recorder and energy meter).

In addition, within the Nobel Grid project, the **Unbundled Smart Meter Concept** has been developed and applied to the existing smart meters. The Unbundled Smart Meter (USM) is a systematization where smart meter functionalities are adequately grouped in two separate (unbundled) components: a) a **Smart Metrology Meter (SMM)** for metrological and hard real-time functions and b) a **Smart Meter Extension (SMX)** that is characterized with high flexibility, so that can support new functionalities and the future evolution of the smart grid and energy services.

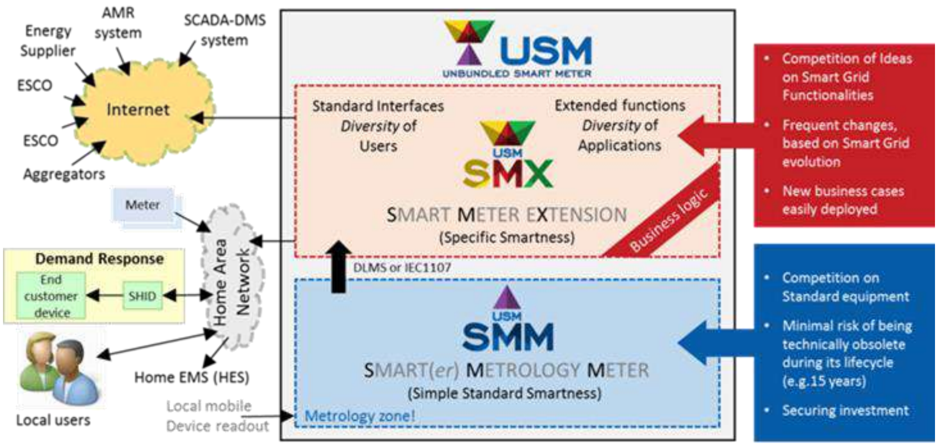

Figure 8. Unbundled Smart Meter Concept

A shown in the figure above, the SMX is responsible for all communication, to both local network (HAN) and public IP-network (internet), as well as running necessary basic applications and third-party applications. The SMs are able to communicate with the SMX for the everyday operations with the distribution power network. They are used by ASM for energy transfer measurements (absorption/consumption) in the context of billing purposes through a DSO's server. This communication framework is depicted in the follow figure.

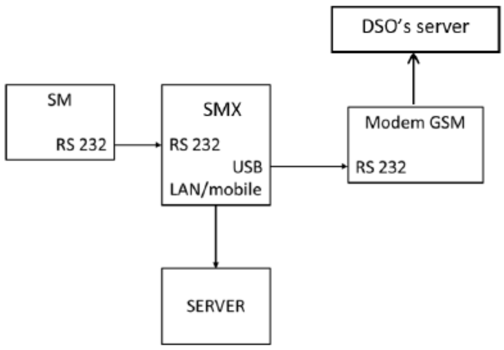

Figure 9. High-level architecture for the communication between SM and SMX

The communication between the electronic smart meter and the GSM modem is realized using a dedicated RS232 channel. The SM is connected to the SMX using the RS232 protocol and the SMX communicates with the modem through the USB gate. The SMX communicates with the server using both LAN and mobile network (3G).

Furthermore, Terni pilot site also includes electric vehicles (EVs) and charging stations provided by EMOT. Two EV charging stations SpotLink EVO and one QC45 quick charging station are deployed in Terni pilot site. SpotLink EVO is characterised by two type 2 sockets, recharging up to 32 A (22 kW) for each socket; SpotLink EVO protection grade is IP54, the impact resistance is IK08 and the protection system is differential type A and type B, with an automatic unlocking of the connector in case of power failure. They are equipped with a single-board computer that allows real-time monitoring and remote management of the charging station such as power output, energy price and remote charging session start/stop. SpotLink EVO connectivity is through modem.

Figure 10. Emotion SpotLink EVO charging stations

Regarding electric vehicles, four Renault ZOE and two Nissan LEAF have been deployed in Terni pilot site.

Renault ZOE, acronym of ZerO Emissions, is a five-door supermini electric car produced by the French manufacturer Renault. The ZOE is powered by a 22 kWh lithium-ion battery pack weighing 275 kg, driving a 65 kW (87 bhp; 88 PS) synchronous electric motor supplied by Continental (the Q210). Maximum torque is 220 N·m (162 lb-ft) with a top speed of 135 km/h (84 mph). The NEDC cycle range is 210 km (130 mi). Renault estimates that in suburban use, the ZOE can achieve around 100 km (62 mi) in cold weather and 150 km (93 mi) in temperate conditions. The car features a charging system called "Caméléon" (Chameleon) charger that allows the ZOE to be charged at any level of power, taking between 30 minutes and nine hours [9].

Figure 11. Renault ZOE customized by Emotion

The Nissan LEAF, acronym of Leading, Environmentally Friendly, Affordable, Family car, is an electric propulsion car introduced by Nissan on the markets in December 2010. It is equipped with an 80 kW (109 hp) synchronous AC electric motor. The first version equipped with a lithium-ion battery, consisting of 48 modules and each of them contains 4 cells for a total of 192, with a capacity of 24 kWh and an autonomy of 199 km NEDC cycle. Since 2016, a 30 kWh Lithium-ion battery with an operating cycle of 250 km NEDC is available as an accessory. Nissan LEAF recharges in alternating current or in direct current. In AC, LEAF uses an on-board charger with a maximum 7.4 kW (32A maximum current, 230V, single-phase) with the Type 2 socket. In DC it uses the CHAdeMO standard up to 50 kW of power. Charging times vary from 5/6 hours to about 7 kW up to 1 hour with direct current charging [10].

Figure 12. Nissan LEAF

EMOT charging stations and electric vehicles send data to EMOT VPS whose details are:

• CPU: 2 core 3.1 GHz;

• HDD: 50 GB;

• RAM: 4 GB;

• S.O.: Ubuntu 16.04 LTS.

Into EMOT VPS run the EV Wrapper Server, OCPP server and API REST.

EMOT charging stations exchange data through a Teltonika RUT230 modem connected to a single-board computer, a Raspberry Pi 3, with a CPU of quad-core ARM Cortex A53 1.2 GHz, a SD of 16 GB, a RAM of 1 GB and a Raspbian Stretch 4.14 S.O.; charging station protocols are Open Charge Point Protocol (OCPP), an application protocol for communication between charging stations and EMOT central management system, and websocket, a computer communications protocol, providing full-duplex communication channels over a single TCP connection. Regarding EV monitoring, EMOT use an on-board diagnostic (OBD) device to retrieve data from the EV; OBD is a IoT component, based on a Raspberry Pi 3 and Carberry; Carberry represents the link between car electronics and Raspberry Pi, which allows the development of end-user applications, such as vehicle diagnostics, data logging, fleet management and tracking. OBD utilize a TCP/IP communication to a TCP/IP server. The network connectivity of the OBD device is via data SIM (UMTS), thanks to a Raspberry module that works as a modem, and the server is a python software; OBD protocol is MQTT and the sampling rate is 5 seconds. The OBD connects to the diagnostic interface from which it is able to extract the information from the electric vehicle control unit using the CAN-bus protocol. The output data format of the OBD is an ASCII string; when the data is sent to the server, it is reorganized into a wrapper, thus obtaining a grouping of the data in JSON format.

The image below describes the EMOT network topology; there are three main networks: the first, top left, is the network to which the EMOT headquarters and charging stations are connected, the second, bottom left, is the network to which the electric vehicles are connected and the third, bottom right, is the OVH network where the EMOT Virtual Private Server (VPS) is hosted.

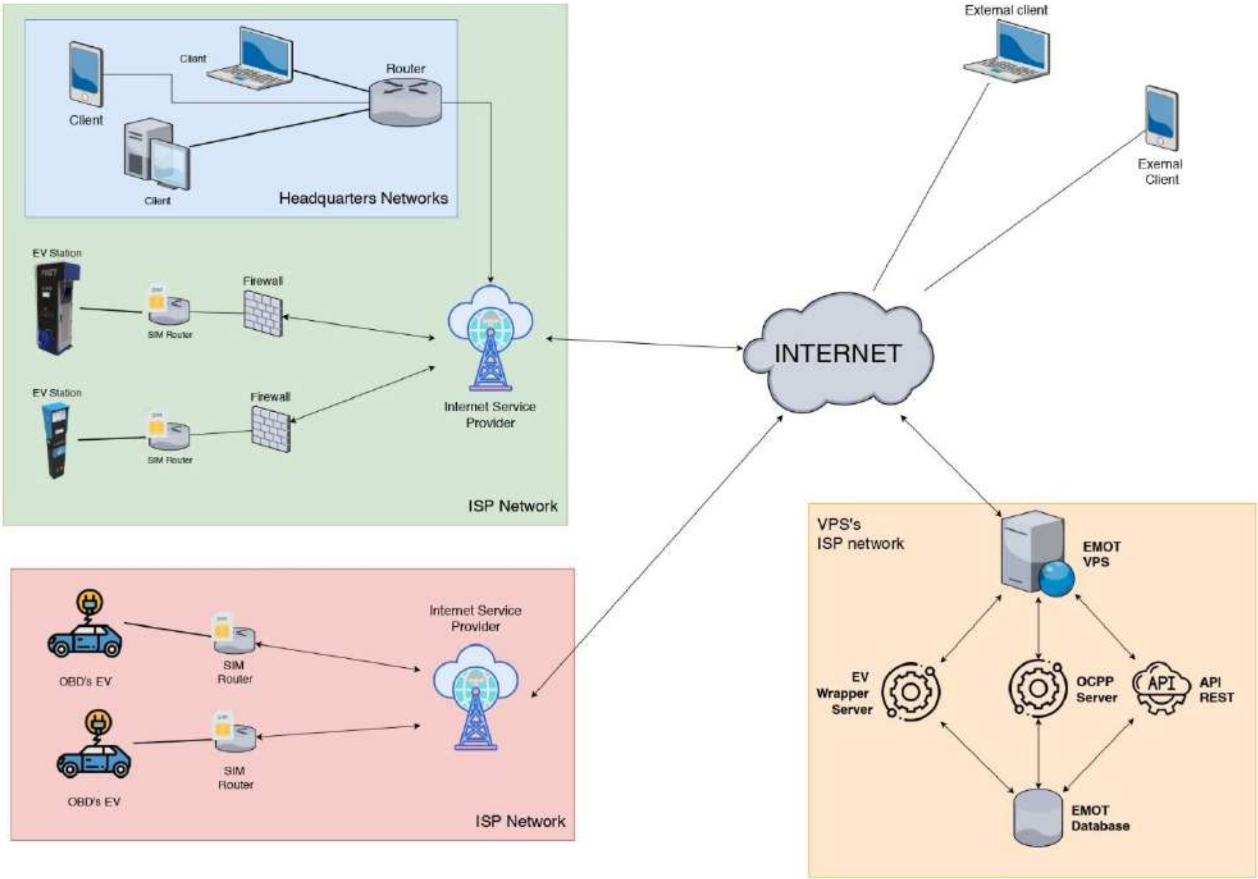

Figure 13. EMOT network topology

Concerning the ASM server-based applications, ASM usually establishes and operates the servers as virtual machines in its server farm. The transit of input data from the SMX to the Enterprise Service Bus is secured by VPN. There are also other security and data protection mechanisms that are in use, including network firewalls built into server farm, secured data transfer (HTTPS) and signed access to server.

3.2 Community-based Virtual Power Plants – KiWi Power

KiWi Power is a UK based aggregator, partner of the National Grid, that aggregates energy assets from multiple sites during peak times to help balance the grid through the use of Demand Side Response programs. The KiWi company maintains active portfolio in all the flexibility programs, including Capacity Market. KiWi Power has developed its own demand response platform, including a proprietary edge hardware (called "Fruit"). This is a metering, communication and control device that is installed at customer side and allows accurate, fast metering and asset control.

During the application of uses proposed in eDREAM, KiWi provides specific tools and components or platform as a whole according to the following:

- **Platform communication via dedicated APIs:** Allow real time data forwarding. Data can be fully anonymized removing all client identification and by adding specific noise in the time series. Data is exchanged in specific formats and transport protocols (e.g. REST JSON API over https to exchange metering data structured using CIM).
- **Device communication:** The edge hardware supports a number of protocols over a variety of interfaces.

The below figure depicts a high-level overview of the components and data flows in a residential estate in Greenwich, where KiWi has installed metering equipment to test residential demand response scenarios.

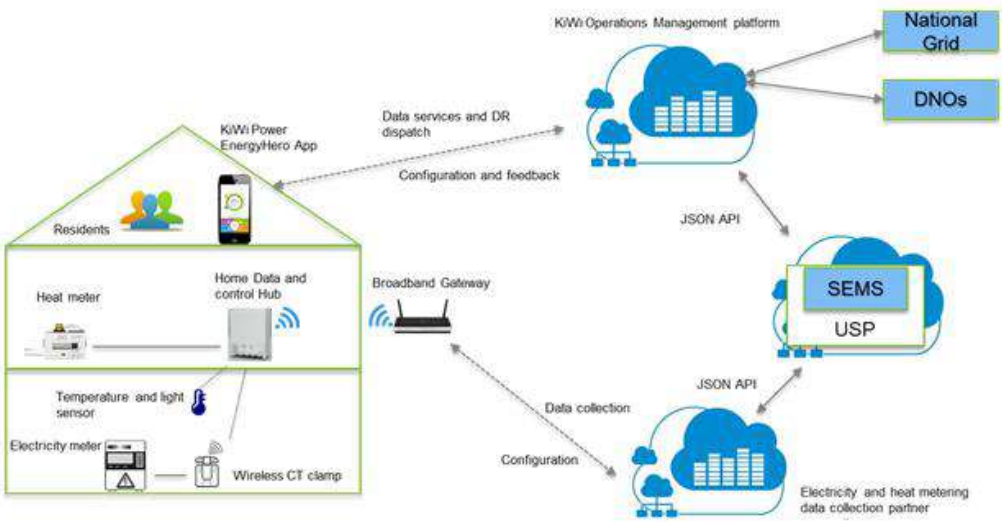

Figure 14. Residential estate in Greenwich for testing residential demand response scenarios

In the context of Demand Side Response programs, the clients receive real time information about their energy consumption, assets status and data analytics through the Client App. The platform receives signals from National Grid / DNOs and sends dispatch signals to Fruit, which in turn enables control strategies on site. It is worth mentioning that for Frequency Response programs, the dispatch is triggered locally on each Fruit controlling unit when grid frequency deviates above certain values.

3.3 Overall Deployment Architecture

This section presents an updated version of the deployment view of the eDREAM platform indicating the interactions with the physical world. The two pilot sites have different technologies of smart meters that will provide the eDREAM platform with the necessary real time measurements for the testing of the architectural components and the tools. These devices are connected to appropriate IoT devices, which will forward the information to the FIWARE Orion Context Broker and to the Off-Chain Data Handlers respectively for the blockchain related information. As a next step, the Context Broker creates different context elements of the received information and manages them. The deployment view of the eDREAM platform is presented in the following Figure 15:

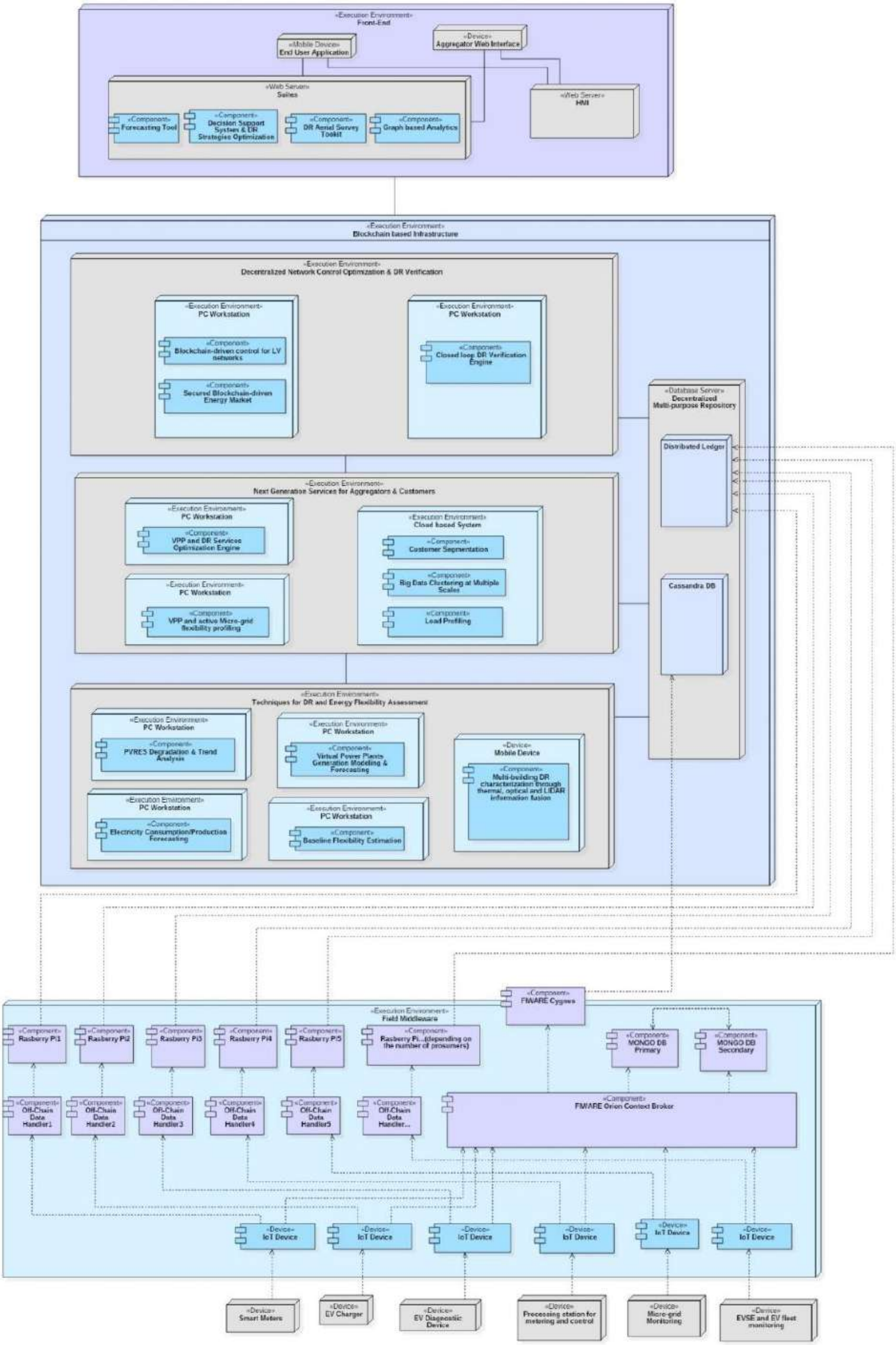

Figure 15. eDREAM Deployment View Architecture

Taking into account the intention for deploying a microservices architecture, we can consider the following options:

- For the integration of the field devices, the combination of IoT Agents with Context Broker can be examined. There are the well-known FIWARE IoT Agents[1] and FIWARE Orion Context Broker[2]. The use of IoT Agents allows the integration and interaction of heterogeneous devices running different protocols (due to the lack of globally accepted standards) that are accessible through multiple wireless technologies. It collects data from devices using heterogeneous protocols and translates them into the standard platform language: NGSI entities (allowing also to send commands to devices). The platform supports several IoT protocols with a modular architecture where the modules are the aforementioned "IoT Agents". In order to select the right IoT Agent to use, the system integrator should determine first which protocol will be used for the connection of the devices. An overview of this concept is presented below:

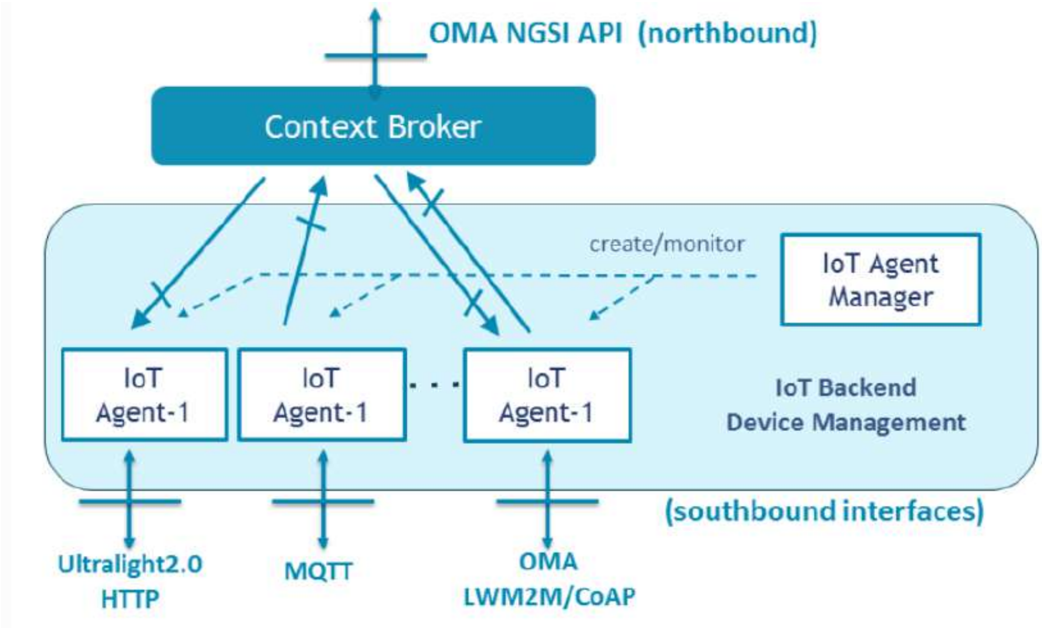

Figure 16. IoT Agents' Concept and Connection

On the other side of the communication, the Orion Context Broker manages context information (e.g. the power consumption provided by a smart meter), enabling to perform updates and brings access to context. It enables to manage context information in a highly decentralized and large-scale manner. This component provides the FIWARE NGSIv2 API which is a simple, but powerful Restful API making possible to perform updates, queries or subscribe to changes on context information.

- The accomplishment of a secure API gateway can significantly reduce coding efforts and make the applications much more efficient, while decreasing errors. A microservices API gateway behaves like any other API gateway. This component provides a front-end layer used to access the below microservices. This gateway creates a single interface for a single application. This means that it can create multiple APIs, one for each platform (e.g. mobile applications, browsers, serve-side applications) that needs to support. Thus, an API gateway can create a custom API for each of these clients so that the client can see just the features it needs. The implementation of the API gateway shall be examined in parallel with the implementation of OpenID connections and identity management. The employment of the OpenID Connect 1.0, that is a simple identity layer on top of the OAuth 2.0 protocol, allows clients to verify the

[1] https://github.com/telefonicaid/iotagent-node-lib
[2] https://github.com/telefonicaid/fiware-orion

identity of the end-user based on the authentication performed by an Authorization Server, as well as to obtain basic profile information about the end-user in an interoperable and REST-like way. OpenID Connect allows all categories of clients, including mobile, web-based and JavaScript clients, to request and receive information about authenticated sessions and end-users. The realization of the API gateway and the OpenID connections will be enabled by ensuring the platform security. Two possible open source technologies, that can be used in order to secure the platform, are the Kong[3] and Keycloak[4]. For both technologies, the Postgresql[5] can be used as a backend database.

- All the microservices shall be able to communicate with each other through rest APIs or AMQP/MQTT messages. A possible solution for message broking is the RabbitMQ[6] that can be easily developed over python or Java using Pika or Java AMQP-client.
- Finally, one more step for the appropriate delivery of microservices is the use of docker[7] concept that is an open source tool for the software product management and orchestration. This tool enables the creation, deployment and running of application by using containers. Containers allow a developer to group an application with all the necessary components, such as libraries and other dependencies and deliver this as one package. The concept of container makes the software independent concerning running on different operating systems regardless of any customized settings that operating machine might have. In addition, the docker as an open source tool can be extended, so as to meet different end-user's needs if additional features are required.

For the testing and implementation of the secure distributed ledger technology, the Parity Ethereum Client[8] is used that is a P2P network and supports the software Solidity V0.4. A testbed is used for exploring Proof-of-Authority capabilities (using Aura validation engine) and Proof-of-Stake concept. The topology of this testbed is shown below in Figure 17:

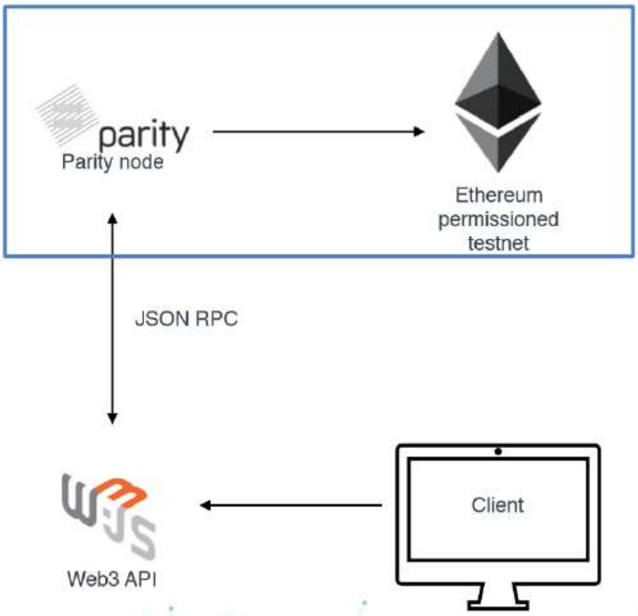

Figure 17. Testbed Topology for the Distributed Ledger Technology

[3] https://konghq.com/kong-community-edition/
[4] https://www.keycloak.org/
[5] https://www.postgresql.org/
[6] https://www.rabbitmq.com/
[7] https://www.docker.com/, https://docs.docker.com/compose/
[8] https://www.parity.io/ethereum/

3.4 Field Devices

In this section, the field devices, that are going to be used in the two pilot sites for taking real time measurements, are presented along with their functionalities, technical characteristics and installation requirements. The format of information is based on the template that has been created for the collection of communication, hardware and software requirements for each device.

Table 77. Smart Metrology Meter (SMM)

Device: Smart Metrology Meter	
Name	Smart Metrology Meter (SMM)
Short Description	Wally-A is used in the pilot as SMM. It is used for monitoring the block of energy units within the pilot as well as to provide power analysis. 
Measurement	Electrical values: § Voltages and Currents § Active, Reactive and Apparent power § Active and Reactive (4 quadrants) Energy § Power Factor § Frequency § Flicker (Pst e Plt) § Voltages and Currents Harmonics and Interharmonics (up to 50° order) § Voltage Unbalance § Voltage Dips and Swells § Voltage Interruptions (short and long) § Rapid Voltage Changes § Waveforms (window records with programmable Pre and Post-Triggers
Digital/Analog Signals	Output 4 – 20 mA
Functionality	Wally-A is a metering device with high level class of precision, certificated according to the standard CEI EN61000-4-30
Physical Characteristics	
Dimensions	Rack 19” Standard – 3U
Weight	3.2 Kg
Material	Plastic and steel
Mounting	Compliant with the rack standard

Hardware Requirements	
Power Requirements	Voltage: 80÷275 Vac/dc - 50/60Hz Power consumptions: 20 VA Battery: 12 V 0.8Ah Pb sealed Battery operating time: 30 minutes, self-limited
Data Connections	Modem type: Internal, Quadband 850/900/1800/1900 MHz Networking: GSM and GPRS/UMTS Internal antenna Optional external antenna SIM holder accessible
Data Format	.csv, .json, .xml, .pqdif.
Data Size	In conjunction with SMX average is 1 MB / day
Data Availability	Curl service HTTP FTP
Transmission Frequency	One transmission every 5 seconds expandable to once a day
Software Requirements (e.g. API creation)	
Software Required	-
Software Details	-

Table 78. Smart Meter Extension (SMX)

Device: Smart Meter Extension	
Name	Smart Meter Extension (SMX)
Short Description	Smart meter extension is a result of Nobel GRID project [11] and it is devoted to create a link between SMM and external world, since it is able to communicate with different protocols (e.g. DLMS, OpenADR, IEC61850) and compliant with different interfaces (e.g. USB, RS-232, RS-485). 
Measurement	All the data collected by the meter
Digital/Analog Signals	Digital signals
Functionality	Data gathering
Physical Characteristics	
Dimensions	15X14X17 (cm)

Weight	0.3 kg
Material	Plastic
Mounting	Compliant with DIN bar
Hardware Requirements	
Power Requirements	0.5 A 5 V DC
Data Connections	3G sim, Ethernet, Internet protocol, Supporting VPN
Data Format	.txt, .json
Data Size	1 MB/day
Data Availability	Real time (5s delay)
Transmission Frequency	Every 5s
Software Requirements (e.g. API creation)	
Software Required	-
Software Details	-

Table 79. EVO Emotion

Device: EVO Emotion	
Name	EVO Emotion
Short Description	Electric Vehicle Supply Equipment 
Measurement	Power, Voltage, Current, Energy
Digital/Analog Signals	-

Functionality	Charge the battery of an electric vehicle. Respond to DR campaigns, starting and stopping remote charging.
Physical Characteristics	
Dimensions	36x28x151 (cm)
Weight	-
Material	-
Mounting	Set on the ground
Hardware Requirements	
Power Requirements	Nominal Current: 32 A Nominal Voltage:230VAC(monophase)/400VAC(triphase)
Data Connections	GSM
Data Format	OCPP
Data Size	Some kBs
Data Availability	Continuous or On Demand
Transmission Frequency	Every 5 s
Software Requirements (e.g. API creation)	
Software Required	*Yes*
Software Details	*Software owned by Emotion*

Table 80. OBD Emotion

Device: OBD Emotion	
Name	*OBD Emotion*
Short Description	*EV on-board diagnostic device* 
Measurement	Battery State-of-Charge (%); Residual Autonomy (Km);

	Needed time to Full Charge (m); Geolocation (geographic coordinates); Doors Car State (Open/Closed); Engine Car State (On/Off).
Digital/Analog Signals	-
Functionality	Collect data from the EV and send data to the server.
Physical Characteristics	
Dimensions	100 x 80 x 30 (mm)
Weight	-
Material	-
Mounting	Inside the electric vehicle
Hardware Requirements	
Power Requirements	Nominal Current: 250 mA Nominal Voltage: 5 V
Data Connections	GSM
Data Format	JSON
Data Size	Some kBs
Data Availability	Periodic
Transmission Frequency	Every 5 s
Software Requirements (e.g. API creation)	
Software Required	*Yes*
Software Details	*Software owned by Emotion*

Table 81. Fruit KiWi Power

Device: Fruit	
Name	Fruit KiWi Power
Short Description	Low-cost processing station designed for Demand Response and Battery Management. The Fruit provides a wide range of metering and control functions and is integrated with the KiWi Operations Management Platform to allow easy set-up and monitoring for any application. It provides flexible interfacing options including GPIO, RS232, RS485 and relay terminals and supports Modbus/TCP and Modbus/RTU.

	
Measurement	Supply voltage (V); Current consumption (at 12V) (mA); Current consumption (at 24V) (mA); Pulse input voltage (V); Pulse measurement frequency (Hz); Maximum number of Segments; Relay terminal voltage (V); Relay terminal current (AC) (A); Relay switching power (VA).
Digital/Analog Signals	-
Functionality	Enabling centrally-dispatched Demand Response programs such as STOR and Capacity Market; Battery Energy Storage control (full EMS); Enabling both static and dynamic Frequency Response programs, such as FCR, FFR and EFR; Monitoring and control of generation assets; Integration with Building Management, SCADA and PLC systems; Environmental monitoring; Metering and sub-metering of electrical installations; DNO constraint management; Remote Telecoms Unit (RTU) for TSO integration.
Physical Characteristics	
Dimensions	114 x 100 (mm)
Weight	-
Material	-
Mounting	Compliant with DIN-EN 60715 TH 35
Hardware Requirements	
Power Requirements	32-bit ARM Cortex R4
Data Connections	WiFi standards: 802.11 a/b/g/n, dual-band 2.4Ghz & 5GHz; Ethernet: 10/100MHz.

Data Format	JSON
Data Size	-
Data Availability	Local data storage capacity without cloud access: Up to 1 year (configuration-dependent); Centralised dispatch latency: < 100ms (communications-dependent); Local frequency dispatch latency: < 40ms.
Transmission Frequency	Maximum Cloud data rate: 10 readings/input/s; RS232/RS485 data rate: 1Mbaud.
Software Requirements (e.g. API creation)	
Software Required	-
Software Details	-

4 Conclusions and future work

This deliverable has presented the current and first validated status of the tools and hardware pilot installations within the eDREAM project framework, drawing close ties with the technical WPs of the DREAM project, that is WP3, WP4, WP5, WP6. These WPs though drawing towards their finalization, are not yet finalized. Thus, the verification process along with its respective results is still ongoing. This is also depicted in the results of this report, as shown in the table below.

Table 82. Overall results

Tool	Integration Status	Development Status	Problems Encountered/Resolver	Performance Status	Improvement Recommendations
Electricity Consumption/Production Forecasting	Under Development	Under Development	Yes	Not yet Available	Yes
PV/RES Degradation Rate & Trend Analysis	Under Development	Final Version	Yes	Yes	Yes
Baseline Flexibility Estimation	Not Applicable	Final Version	Yes	Not yet Available	Yes
Virtual Power Plants Generation Modeling & Forecasting	Under Development	Under Development	Yes	Not yet Available	Yes
Multi-Building DR Characterization through thermal, Optical and LIDAR information fusion	Under Development	Under Development	None reported	Not yet Available	None reported
Load Profiling	Finalized	Final Version	None reported	Yes	Yes
Big Data Clustering at Multiple Scales	Finalized	Final Version	None reported	Yes	None reported
Customer Segmentation	Finalized	Final Version	None reported	Yes	None reported
VPP & DR Services Optimization Engine	Under Development	Under Development	None reported	Not yet Available	None reported
Distributed Ledger	Finalized	Final Version	None reported	Yes	Yes
Blockchain-driven control LV networks	Under Development	Final Version	None reported	Not yet available	None reported
Secured Blockchain-driven Energy Market	Under Development	Final Version	None reported	Yes	None reported
Closed-loop DR Verification Engine	Finalized	Final Version	None reported	Not yet available	None reported
Graph-based Analytics-HMIs	Under development	Under development	Yes	Yes	Yes
DSS & DR Strategies Optimization	Under development	Under development	None reported	Not yet available	None reported
DR Aerial Survey toolkit	Under development	Under development	Yes	Yes	Yes
Forecasting Tool	Finalized	Final Version	None reported	Not yet available	None Reported

From Table 82 above most of the tools seem to be in a mature state and from the already collected information seem to comply with the guidelines set during the activities of WP2. It should be stated however that the WP2 activities are still underway, therefore the user requirements, functional requirements and system specifications, are constantly being updated to match the most state-of-the-art technologies and end-user demands regarding DR solutions for every energy related stakeholder involved in the eDREAM ecosystem. This, in turn, affects the ongoing development of the tools. Regarding the field device equipment deployment is currently well under way. More details and information are to be expected in the final version of this document, and hence this report is to be considered a live document regarding the technology verification process.

References

[1] eDREAM Deliverable D2.5 – Requirement-Driven System Development V2. June 2019

[2] eDREAM Deliverable D7.1 – Validation Plan V1. May 2019

[3] eDREAM Deliverable D3.2 – Recommendations for baseline load calculations in DR programs V1. April 2019

[4] eDREAM Deliverable D3.6 – Recommnedations fot baseline load calculations in DR programs V2. May 2020

[5] eDREAM Deliverable D3.4 – Aerial 3D models and simulation procedures for DR estimation V1. May 2019

[6] eDREAM Deliverbale D4.1 – Specification for Improved Decision Making and DR Optimization toolsets V1. April 2019

[7] eDREAM Deliverable D5.1 – Blockchain platform for secure and distributed management of DR programs V1. May 2019

[8] eDREAM Deliverable D5.3 – Consensus based Techniques for DR validation and financial settlement.

[9] https://www.car.info/en-se/renault/zoe/zoe-2018-16314878/specs

[10] https://www.car.info/en-se/nissan/leaf/leaf-86616/specs

[11] https://nobelgrid.eu/